

Development of an Open Source Prosthetic Hand Platform

A Thesis

Presented to the Faculty of
California Polytechnic State University,
San Luis Obispo

In partial fulfillment of the requirements for the degree
Master of Science in Biomedical Engineering

By

Scott Garrett

June 2011

© 2011

SCOTT GARRETT

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Development of an Open Source Prosthetic Hand Platform

AUTHOR: Scott Garrett

DATE SUBMITTED: June 2011

COMMITTEE CHAIR: Dr. Lily Laiho

COMMITTEE MEMBER: Dr. Tom Mase

COMMITTEE MEMBER: Dr. Robert Szlavik

ABSTRACT

Development of an Open Source Prosthetic Hand Platform

Scott Garrett

In the field of upper extremity prosthetic devices, advancements in technology drive the design of products which are becoming capable of restoring the lost functions of the native hand. While several dexterous devices have been developed to serve this purpose, they remain prohibitively expensive and thus are not a viable option for many upper extremity amputees. To address this problem a prosthetic hand platform was developed utilizing the open source Arduino microcontroller and off-the-shelf electrical components. Using these resources, a novel finger actuation mechanism was developed to show how a prosthetic hand platform could be developed which is capable of individual finger actuation, multiple actuation modes, sensing of forces at the individual fingers, providing force feedback to the user, and control of finger actuation through a variety of control inputs.

After going through several iterations of hand's mechanical components, electronics, and firmware a final prototype was built to showcase the possible capabilities of the open source prosthetic hand platform. This prototype consisted of several groups of subcomponents including an auto-flexing / extending finger design, a modular palm/ servo attachment base, and a wrist section which housed the hand's electronic components, power supplies, force feedback system.

The open source prosthetic hand platform was then verified using a series of tests to quantify several performance characteristics of the final prototype. Battery life and grip strength during continuous use were evaluated and demonstrated that the hand could provide consistent grip force during up to two hours of initial continuous use. Also, the grip performance of the hand was assessed through the grasping of spherical objects with varying surface textures, diameter, and weight. Furthermore the hand was tested in various “real life” applications including manipulating and sorting small objects, opening doors, grasping moderately heavy objects such as water bottles, and sensitive objects such as an egg. Lastly, the platform was connected to a myoelectric input circuit to demonstrate compatibility with advanced electro-physical inputs. These tests demonstrated that the platform was capable of performing some of the dexterous tasks performed by prohibitively expensive available robotic upper extremity prosthetic devices.

Further developments could be made to the open source prosthetic hand platform including enhancements to the platform’s finger force sensing and feedback mechanisms, consolidation of the electronics, refinement of the auto-flexing / extending fingers, and integration with a silicone covering and patient’s residual limb socket. These future iterations of this platform could help provide a dexterous prosthetic hand platform at lower cost to a wider patient base.

ACKNOWLEDGEMENTS

Special thanks to the Cal Poly San Luis Obispo Biomedical and General Engineering Department for its significant support. The department has provided the technical training and access to hands on laboratory and manufacturing experiences which made the development of the open source prosthetic hand platform possible. I would like to thank my thesis advisor Dr. Lily Laiho in particular for her helpful guidance and assistance throughout the design and verification of this project.

I would also like to express my gratitude to the Quality of Life Plus (QL+) program which has been an invaluable source of inspiration and assistance in the development of the open source prosthetic hand project. QL+ advisors and collaborators, including Dr. Tom Mase, Matt Greibel, and Nick Butler, have provided key insights in addition to alternate applications of the technologies developed for this platform. By demonstrating the needs of the real patients, this lab and its researchers were among the strongest motivators to initiate this project.

Lastly, I would like to express my sincere gratitude to the Hannah Forbes Senior Project Fund which provided the funding for development of this project. This funding avenue provided the means to pursue a novel research venue which I hope will benefit the field of robotic prosthetic design as a whole.

Table of Contents

LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF NOMENCLATURE	xiii
1.0 INTRODUCTION	2
The Need for Advanced Upper Extremity Prosthetics	2
2.0 BACKGROUND	3
Prosthetic Hand Designs	3
Microcontrollers	9
The Arduino Platform	10
Integrating Sensors and Actuators with Microcontrollers.....	12
Force Sensitive Resistors	13
Servos	16
Myoelectric Amplification Circuitry	17
Myoelectric Circuitry at Cal Poly San Luis Obispo	21
Prosthetic Hand Designs Through Cal Poly and Quality of Life Plus	22
3.0 DESIGN PROCESS	24
Goals	24
Functional & Non-Functional Requirements	24
Desired Design Features.....	25
Design Assumptions	25
Project Timeline - Proposed and Actual	26
Proof of Concept Prototype	27
Servos	29
Force Sensitive Resistor	30
Button Case Change	31
Current Sensing Module	32
Two Axis Accelerometer	33
Myoelectric Input Integration	34
Open Hand 1.1 Code (See OpenHand I.0)	35
Open Hand 1.2 Code.....	36
Rev. 1 Mechanical Design	37
Rev. 2 Mechanical Design	38
Rev. 3 Mechanical Design	40
Rev 3 Mechanical Design (Thumb).....	41
4.0 FINAL DESIGN	43
Final Prototype	43
Component List and Prices	50
5.0 TESTING AND RESULTS.....	53
Power Consumption Assessment	53
Spherical Object Texture and Weight Effects on Grip Test	58
Limited Real-World Use Assessment.....	62

Myoelectric Input Functional Test	63
6.0 CONCLUSIONS	64
WORKS CITED	68
APPENDIX A: CODE	70
ServoRock.....	70
ServoRock2.....	71
FSR	71
CurrentSensingModule	72
Accelerometer Sensor.....	72
APPENDIX B: CODE	74
Proof of concept prototype Code:.....	74
Open Hand 1.1.0 Code:	77
Open Hand 1.1.1 Code:	79
Open Hand 1.2.1 Code:	81
Open Hand 1.3.1 Code:	83
APPENDIX C: Final Prototype Engineering Drawings.....	88

LIST OF FIGURES

Figure 1 A Body Powered Prosthetic (the Physionetics V2P Prehensor) which utilizes a shoulder harness to actuate the prehensor claw (Physionetics LLC)	4
Figure 2 The Otto Bock - SensorHand Speed, An Electric prosthetic hand demonstrating close anthropomorphic similarity to a patients native hand (Otto Bock)	5
Figure 3 One of the anthropometric grips of the Touch Bionics I-LIMB, an electronic prosthetic hand (Touch Bionics).....	6
Figure 4 Touch Bionics shown performing the power grip in two orientations (Touch Bionics)	7
Figure 5 The DEKA "Luke" arm, a highly dexterous upper extremity electronic prosthetic being used by a patient with an above the elbow amputation (DEKA Research and Development corporation)	8
Figure 6 Arduino microcontroller boards showing the variety of form factors and IO capabilities of the platform. A: Arduino Uno, B: Arduino Mini, C : Arduino Mega, D: Arduino Lily Pad (Arduino)	10
Figure 7 Arduino programing interface used to compose code or "sketches", upload code to the Arduino microcontroller, and monitor serial communication	11
Figure 8 Force sensitive resistor assembly showing the thin film layers which comprise the sensor. The base layer includes an array of electrodes which when compressed through spacer opening to the top printed semiconductor layer, decreases the resistance of the FSR [12].....	13
Figure 9 Plots showing the resistance (A), and conductance (B) measurements in response to a load applied to a force sensitive resistor. These sensory inputs can be utilized by a microcontroller to determine the pressures applied at the site of the sensor[12]	15
Figure 10 Two servos with different weights and torque outputs. A: 52.1g with 83.32oz/in torque, B: 5.8g with 8.4oz/in torque	16
Figure 11 Myoelectric circuit showing a subjects muscle contraction being detected and used as a control input for a servo (Electrical components of the PolyGrasp Hand developed by Nickolas Butler, Matt Greibel, and Max Maloney).....	18
Figure 12 Diagram of the electrode placement for surface for surface electromyography (sEMG).....	19
Figure 13 Unfiltered EMG signal showing the high levels of noise caused by adjacent electrical activity in bodily tissues and interference from common electrical devices	20
Figure 14 1000 point averaging windows of the full-wave rectified version of the EMG signal.....	21
Figure 16 The prosthetic hand developed for a navy SEAL through the QL+ program.....	23

Figure 17 proof of concept prototype developed to demonstrate independent servos control and multiple operational modes using the arduino microcontroller	28
Figure 18 Schematic of servo integration with the Arduino “Stamp” Mini	30
Figure 19 Schematic of force sensitive resistor integration with the Arduino “Stamp” Mini	31
Figure 20 Schematic of momentary switch integration with the Arduino “Stamp” Mini	32
Figure 21 Schematic of current sensor integration with the Arduino “Stamp” Mini	33
Figure 22 Schematic of two axis accelerometer integration with the Arduino “Stamp” Mini	34
Figure 23 A block diagram of the Open Hand 1.1 code with the effects of the input electronic components (green) on the Arduino’s programmed algorithms (yellow) and their resulting effects on the hand’s outputs (red). The Open Hand 1.1 code features 4 operational modes which affect the functions of the hand’s fingers and are toggled between using a mode button	36
Figure 24 A block diagram of the Open Hand 1.2 code with the effects of the input electronic components (green) on the Arduino’s programmed algorithms (yellow) and their resulting effects on the hand’s outputs (red). The Open Hand 1.2 code features 2 operational modes which affect the functions of the hand’s fingers and are toggled between using a switch which detects when the thumb is in line with the index finger.....	37
Figure 25 Rev. 1 mechanical design showing the arrangement of three servos used to power a five-fingered version of the prosthetic hand	38
Figure 26 A finger design constructed from bent sheet metal with internal lever arms which cause bending of the distal portions of the finger	39
Figure 27 Auto-grip motion of the fingers caused by internal lever arms.....	39
Figure 28 A simplified finger bending mechanism which utilizes juxtaposed levers that also function as structural elements. These levers allow for the distal tip of the finger to curl inward as the two levers are flexed toward the palm.....	41
Figure 29 A switch located in the thumb housing which signals the position of the thumb to the hand's microcontroller.....	41
Figure 30 The final prototype of the open source prosthetic hand platform featuring three independently actuated fingers and a thumb which can be utilized to toggle between grasping and pinching modes of operation	43
Figure 31 The palm portion of the open source prosthetic hand which utilizes a modular finger attachment mechanism in order to create an anthropometric grip	44
Figure 32 The positioning of a force sensitive resistor on the distal portion of the fingers in order to measure grip pressure at the fingertip	45
Figure 33 The push pull rods used to open and close the hands fingers.....	45
Figure 34 The final prosthetic hand prototype showing the 30 degree incline at the wrist to match the neutral positioning of a patients native hand	46

Figure 35 A vibrating motor located in the wrist portion of the prosthetic hand used to provide force feed back to the user.....	47
Figure 36 The open and closed grasp position (A&B) and the open and closed pinch position (C&D) of the final prototype hand	48
Figure 37 A schematic of the electronics utilized in the final prototype. The hand features 4 FSRs which are used as inputs for the hand's 3 servos causing finger actuation and force feedback from a vibrating motor. The optional connections for the microcontroller's USB interface is also shown.....	49
Figure 38 The setup used to measure battery drain using a multimeter and grip force using a hand dynamometer during the continuous use test	54
Figure 39 Attachment of an adapter bar to the hand using zip ties in order to align the force transducer in line with the grasp of the fingers.....	55
Figure 40 Connection of the multimeter to break out leads used to measure the potential across the servo power supply.....	55
Figure 41 Decrease in servo power supply voltage during continuous cyclic contraction of the hand over 2 hours	56
Figure 42 Decrease in grip force during continuous cyclic contraction of the hand over 2 hours	57
Figure 43 spherical objects used to characterize the hand's lifting properties: a Baseball, roller hockey ball, tennis ball, golf ball, and ping pong ball	59
Figure 44 The lifting net loaded with a known mass attached to a spherical object used to characterize the hand's lifting properties. The purpose of the net was to increase the load lifted with each ball without changing the object's outer diameter or texture.	60
Figure 45 Lifting capacity of the hand with respect to each object's weight and diameter. The hand was capable of lifting greater total loads as the diameter of the object decreased	61
Figure 46 Lifting of both heavy and delicate objects using the prosthetic hand's "grasp" and pinch modes	62
Figure 47 Myoelectric circuit configured to read muscle action potentials in the arm	63

LIST OF TABLES

Table I Timetable of thesis tasks with proposed and actual completion dates ...	27
Table II List of components that were used to construct the open source prosthesis hand's proof of concept prototype.....	28
Table III Operational modes designed into the firmware of the proof of concept prototype hand	29
Table IV Bill of materials detailing all purchase orders associated with the development of the open source prosthetic hand platform including materials not necessarily included in the final prototype	51
Table V Bill of materials detailing only materials and components included in the final prototype.....	52

LIST OF NOMENCLATURE

QL+	Quality of Life Plus
IO	Input/output
FSR	Force sensitive resistor
sEMG	surface electromyography
RC	Remote control
LED	Light emitting diode
PWM	Pulse width modulation
CNC	Computer numeric control
SOC	State of charge
A to D	Analog to digital

1.0 INTRODUCTION

The Need for Advanced Upper Extremity Prosthetics

Both unilateral and bilateral upper extremity amputations are procedures which ultimately leave patients with significant functional deficits. In addition to treating congenital defects, amputations are also performed in response to traumatic injury, or tumor growth. Traumatic injury represents the majority of these amputations [1]. Despite the difficulties which result from the loss of a limb, many amputees do not utilize prosthetic devices to aid them in daily activities. Prosthetic devices are commonly rejected by their users due to pain, poor comfort, and lack of functionality [1]. Furthermore in the case of severe injuries, the use of an upper extremity prosthetic device may be inhibited by the formation of pressure ulcerations [2]. Consequently there is a need for more advanced upper extremity devices which better restore the functionality of a native hand while also properly integrating with the patient's residual limbs.

2.0 BACKGROUND

Prosthetic Hand Designs

The design of prosthetic hands is a process which balances the importance of anthropometric appearance with achieving the best possible dexterity and function. The importance of these respective concerns often varies between users depending on their individual needs. Furthermore upper arm prosthetics generally have low usage as a result of the difficulty of coordinating reach and grasp motions [3]. Due to this complication there is great interest in developing prosthetics which better address this functional deficit. Most conventional prosthesis feature only a few of the human arm's 22 degrees of freedom including opening and closing of the hand, flexion and extension of the elbow, and rotation of the wrist. [4]. The two primary classes of prosthetic hand are mechanical or body powered and electric prosthetics. Body powered prosthetics usually are operated using a shoulder harness or similar cable driven system attached to the patient's abdomen (Figure1).



Figure 1 A Body Powered Prosthetic (the Physionetics V2P Prehensor) which utilizes a shoulder harness to actuate the prehensor claw (Physionetics LLC)

By extending the residual limb away from the body this cable system is put in tension causing the prosthetic hand to open, which is often pretensioned with elastic bands or other energy storing elements. Accordingly flexion of the arm back to the body will ease the tension in the cable system causing the hand to close. Body powered hands are durable, provide sensory feedback, and can provide both high grip and high dexterity functions. Furthermore, they are generally considered lightweight and are much less expensive than electric prosthetic hands, however, they require more gross movement of the body in order to be actuated and are often less aesthetically pleasing [5]. Electric prosthetic hands are operated either through myoelectric circuitry or through manual switches toggled by bony protrusions located in the cuff attaching the prosthetic. Directly interfacing with a patient's residual nerve function can allow for myoelectric signals to be utilized to control the functions of an electric prosthetic and force feedback can be relayed to the patient in the form of

electrical surface stimulation of the limb [6]. These prosthetics are less durable, but can provide sensory feedback and are also capable of advanced dexterous function and strong grip force. Electronic prosthetic hands are generally heavier than body operated prosthetic hands due to the added electronic components and batteries. However operation of these hands is often less strenuous because they are not dependent on gross movement of the shoulders or abdomen for actuation. Lastly electronic prosthetic hands are often more aesthetically pleasing by incorporating anthropometric form factors which more closely mimic the user's native anatomy

(Figure 2).



Figure 2 The Otto Bock - SensorHand Speed, An Electric prosthetic hand demonstrating close anthropomorphic similarity to a patient's native hand (Otto Bock)

State of the Art Electric Prosthetic Hands: Touch Bionics i-LIMB

One advanced upper extremity electronic prosthetic that is currently available is the Touch Bionics i-LIMB. This prosthetic device features five electric motor driven digits which allow the device to conform to various grip patterns [7]. This prosthetic hand can utilize a myoelectric input source which senses contraction of muscles in the patients residual limb to open and close the hand. The i-LIMB has a highly anthropometric design which closely resembles the anatomy of the user's native hand (Figure 3).

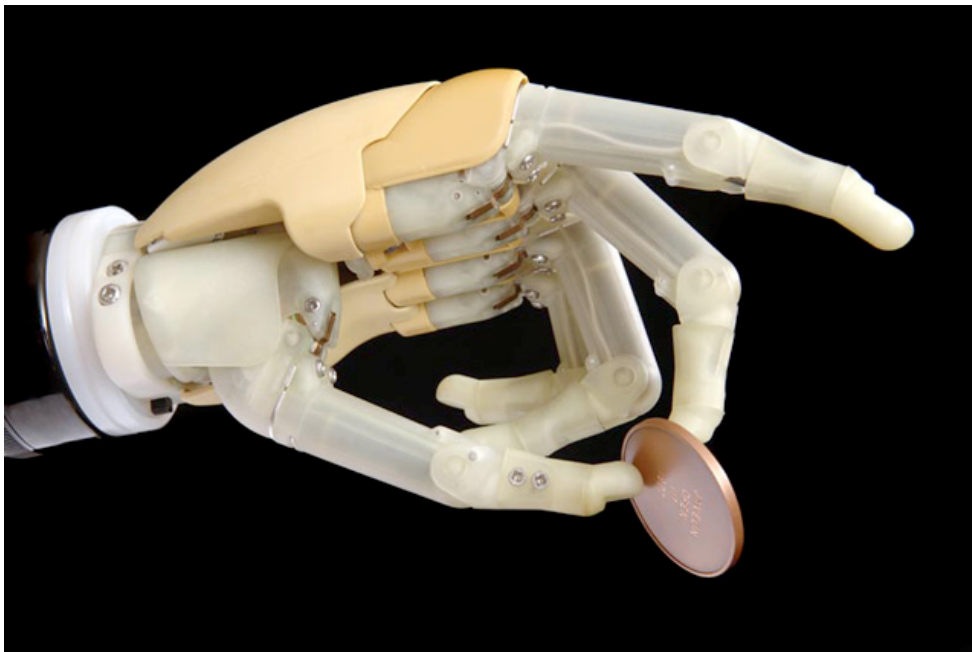


Figure 3 One of the anthropometric grips of the Touch Bionics i-LIMB, an electronic prosthetic hand (Touch Bionics)

The individual fingers can be removed for easy servicing or replaced by the patient's prosthetist. Rotation of the thumb enables several different grip patterns which allows the finger motions to be coordinated for different tasks [8] (Figure 4).



Figure 4 Touch Bionics shown performing the power grip in two orientations (Touch Bionics)

In the key grip the thumb closes on the index finger to pick up small objects (Figure 3). In the power grip all fingers close simultaneously to grip larger objects. In the precision grip either the index or middle finger meets the thumb to perform fine control tasks. Lastly, the index point closes all fingers and extends the index finger in order to push buttons. Additional features including moving the thumb inward to more easily put on jackets has been considered for future software adjustments. The i-LIMB can be configured with either a translucent skin which gives the hand a robotic appearance or with a natural appearance which can be matched to the patients skin tone. Considered one of the most highly functional prosthetic hands on the market the i-LIMB costs approximately \$18,000 before insurance reimbursement (circa 2011).

State of the Art Electric Prosthetic Hands: Deka Luke hand

The DEKA “Luke” arm is a prosthetic arm under development which is being funded by the Defense Advanced Research Projects Agency (DARPA). The DEKA arm is considered one of the most advanced upper extremity prosthetic devices under development due to the arm’s dexterity and multiple degrees of actuation. The arm can be configured to an individual’s particular level of amputation and offers up to 18 degrees of freedom including a powered shoulder, elbow, wrist, and multi-grasp hand [9] (Figure 5).



Figure 5 The DEKA "Luke" arm, a highly dexterous upper extremity electronic prosthetic being used by a patient with an above the elbow amputation (DEKA Research and Development corporation)

Currently the DEKA arm is undergoing clinical trials in which participants control the arm using pressure sensors located in the insoles of their shoe. The arm has also been tested using myoelectric circuitry which has been configured to read

the action potentials conducted by nerves to the patient's muscles in their residual limbs. Furthermore other control inputs are also being researched including neural interfaces placed outside, or even inside, the body [9]. While the functionality of the DEKA arm represents a significant advancement in upper arm prosthetics the device will likely be outside the affordable range of many users with a target price approaching \$100,000.

Microcontrollers

Microcontrollers are small computers used control automated functions of various complex electronic devices. They contain a processing core, limited memory, and input/output circuitry which allow for them to be connected to a variety of electrical components. Microcontrollers are generally utilized for embedded applications such as consumer devices where they are used to coordinate the functions of a device's electrical components. In order to conduct these functions, microcontrollers are loaded with small programs which are either written in custom languages or general purpose languages such as C. Popular microcontroller platforms include, the ARM, PIC, and Atmel microcontrollers which vary in terms of on board capabilities, and programming environments. The Arduino microcontroller platform is an open-source Atmel-based microcontroller which utilizes a highly user friendly C-based programming environment. The open source community supporting this platform also provides a comprehensive source of tutorials and learning guides which make it an ideal platform for developing embedded devices with minimal previous electronics

experience. Because of this, the Arduino microcontroller platform was utilized in the development of the open source prosthetic hand.

The Arduino Platform

The Arduino microcontroller platform is a user friendly prototyping platform which includes open source hardware and software [10]. The microcontroller boards which comprise the hardware of the Arduino platform can be built by users or purchased pre-assembled. Boards available for purchase vary in terms of form factor, connectivity, memory, and the number of digital and analog input/output ports (Figure 6).

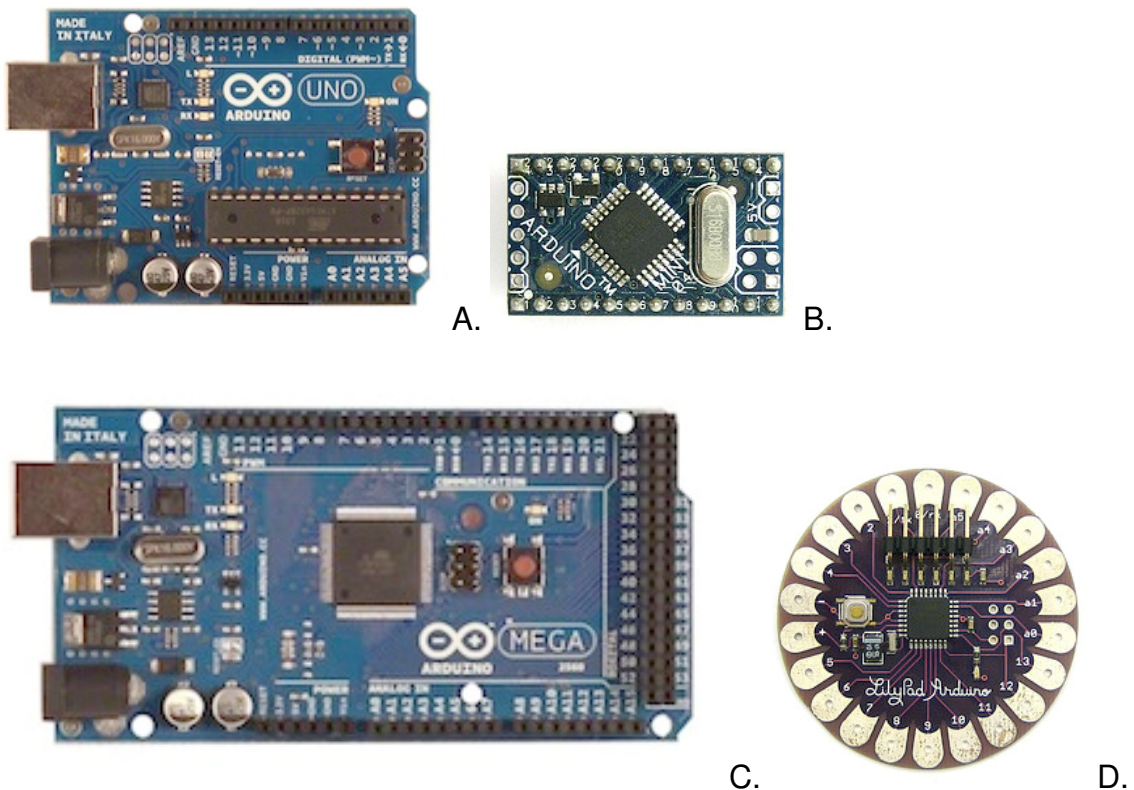


Figure 6 Arduino microcontroller boards showing the variety of form factors and IO capabilities of the platform. A: Arduino Uno, B: Arduino Mini, C : Arduino Mega, D: Arduino Lily Pad (Arduino)

Boards can be connected to a computer for programming via USB directly or through an adapter board with some boards also featuring a serial port connection. Arduino boards can be powered via USB in applications that draw low current. Alternatively a 6-20V power supply (this specification varies by board) can be used to power an Arduino board such as a 9V battery. In high current draw applications it is important that a separate power source be used and connected to a common ground to prevent “brown out” or under powering of the Arduino’s processor. Schematics of the available boards are available online and can be modified to fit specific applications. The software environment used to program Arduino microcontrollers (based on Wiring) is also available free to download online. This environment provides an interface to write and test code prior to uploading to an Arduino board (Figure 7).

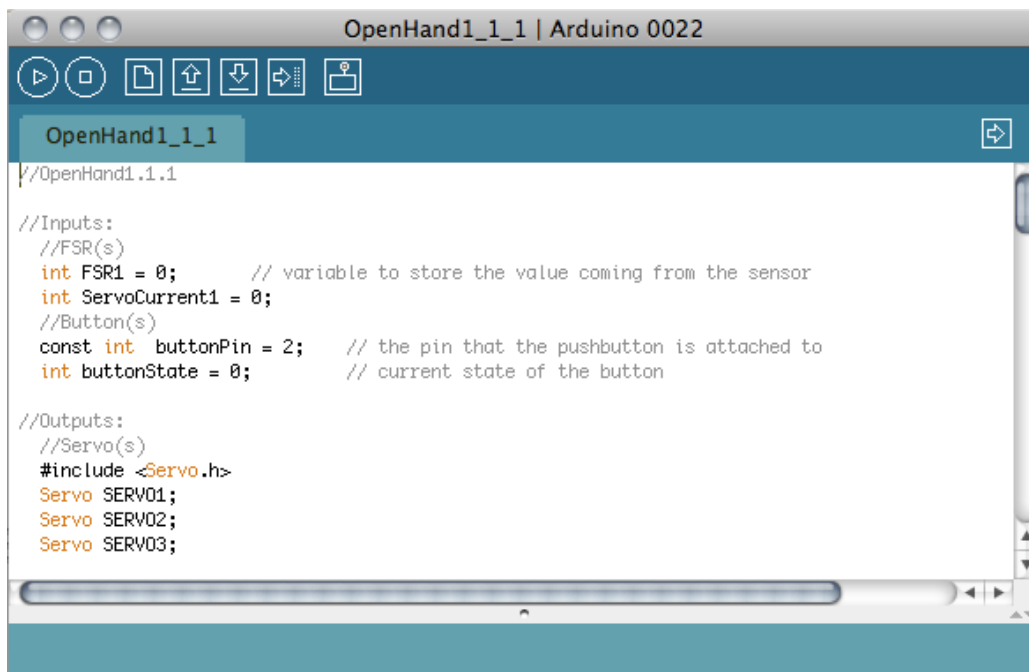


Figure 7 Arduino programing interface used to compose code or "sketches", upload code to the Arduino microcontroller, and monitor serial communication

Once a set of code has been successfully compiled the coding environment can be used to configure the USB or serial connection of the Arduino microcontroller and upload the code. The interface can also be used when the Arduino is connected to the computer either by serial interface or Bluetooth to communicate with the board. Code generally begins with the declaration of variables and I/O ports, followed by a “setup” loop which executes once followed by a “void” loop which will execute for the duration of the Arduino’s operation until the board is reset or the power is cycled. This platform is capable of taking readings from multiple sensor types and integrating these signals with its programmed algorithms to control various mechanical and electronic outputs. Extensive tutorials and examples in addition to user supported forums are available at the Arduino website: arduino.cc [10]. These examples provide information regarding programming syntax in the Arduino programming environment in addition to sample code demonstrating how to interface with various external electrical components [11].

Integrating Sensors and Actuators with Microcontrollers

There are a variety of sensors and actuators which can be configured to perform a variety functions. Sensors can be designed to measure a variety of environmental and physical phenomena using various electrical components. Changes in these phenomena usually results in a change in the electrical properties of a sensor such as its resistance, voltage, output, or digital pulse width output. Using a microcontroller’s analog or digital inputs, these changes can be detected, conditioned, and interpolated to initiate various functions of a

microcontroller's programming. Some of the actions initiated by the sensory inputs of a microcontroller include the activation of actuators. Actuators include a variety of devices which convert electrical power into motion of mechanical components and include devices such as motors, servos, stepper motors, linear actuators, etc. A variety of sensors and actuators were researched and incorporated into the open source prosthetic hand in order to improve the functionality of the device.

Force Sensitive Resistors

Force Sensitive Resistors (FSR) are sensors which decrease in resistance when a force is applied to their active surface [12]. In the case of those utilized in this thesis, the FSR's used were constructed as polymer thin film devices (Figure 8).

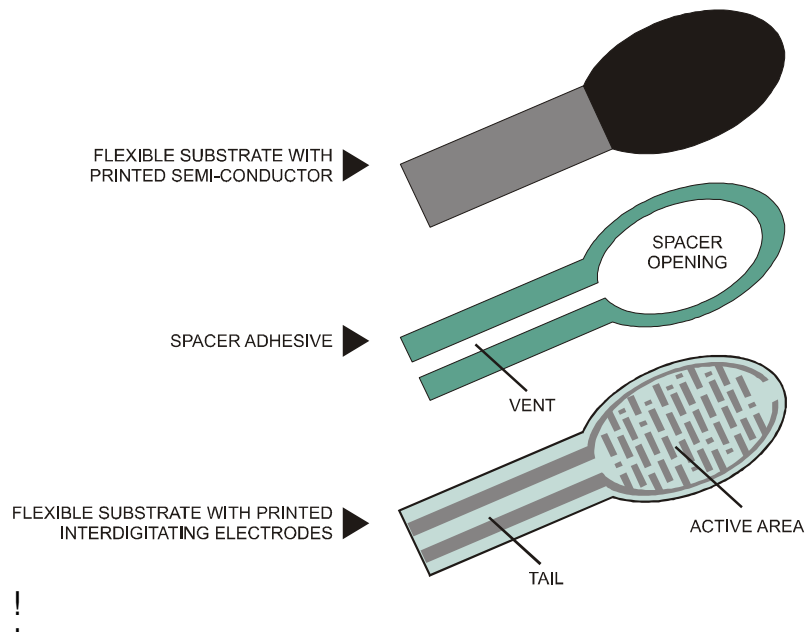
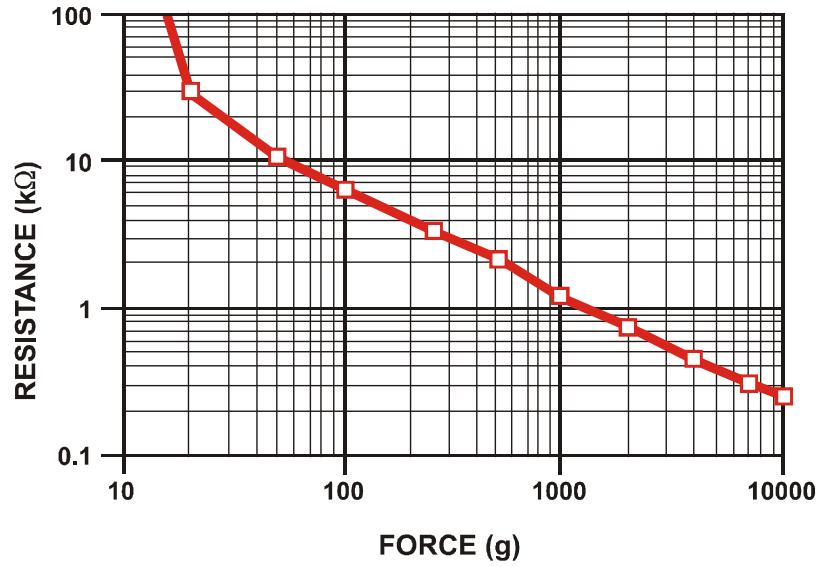
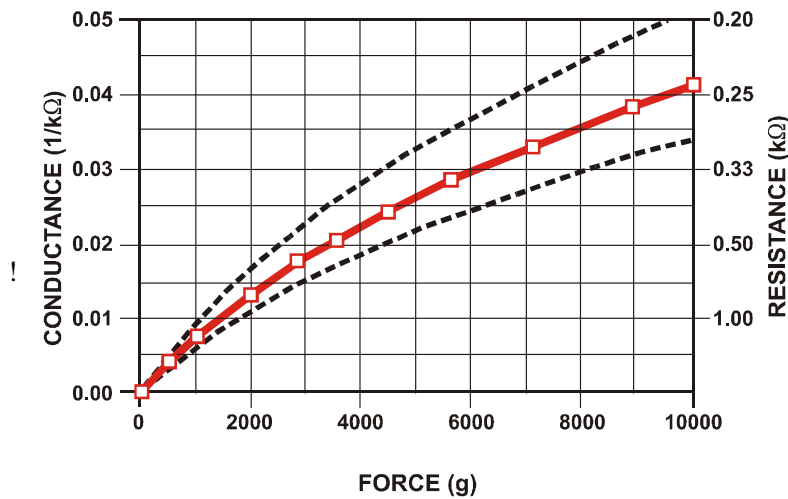


Figure 8 Force sensitive resistor assembly showing the thin film layers which comprise the sensor. The base layer includes an array of electrodes which when compressed through spacer opening to the top printed semi-conductor layer, decreases the resistance of the FSR [12]

While FSR's display similar performance characteristics to strain gauges and load cells they are not considered appropriate for precision measurements due to the noise of their output signal. However, with $\pm 5\%$ accuracy of the FSR's force measurements with a range of up to 20 lbs these devices are useful in the control of electronic devices through a range of forces achievable through human touch [12].



A.



B.

Figure 9 Plots showing the resistance (A), and conductance (B) measurements in response to a load applied to a force sensitive resistor. These sensory inputs can be utilized by a microcontroller to determine the pressures applied at the site of the sensor[12]

Figure 9 shows, the decrease in resistance approximately follows an inverse power-law characteristic ($1/R$) [12]. When the conductance or inverse of the resistance is calculated, a more linear plot is produced which is useful for making approximate measurements and for calibration of the FSR. The FSR's manufacturer, Interlink Electronics, notes that the accuracy of the FSR can vary

between 5-25% depending of the consistency of the measuring procedure involved in the FSR's use.

Servos

Servos are electronic actuators which move according to an analog or digital signal input. They are comprised of a gear box and lever arm which rotates with angular specificity between a 60 and 180 degree range. Servos can be modified to rotate continuously beyond 360 degrees though this affects their positioning specificity. Compared to conventional motors which can generally only be configured to with specific rotational speed and acceleration, servos are desirable in many robotic applications where motion and position specificity are important performance metrics [14]. Off-the-shelf servos come in a variety of torque capacities with size and power requirements generally increasing with torque output (Figure 10).



A.



B.

Figure 10 Two servos with different weights and torque outputs. A: 52.1g with 83.32oz/in torque, B: 5.8g with 8.4oz/in torque

Other factors which affect servo design include gearing ratios, gear materials and casing treatments including water and dust proofing. Some servos can be geared to improve their mechanical advantage and improve torque, however, this normally comes at the expense of the servo's rotational velocity. Most servos are constructed with hard plastic gear sets although titanium and other metal gear servos are available which are more resilient in high wear applications. Some of these metal gear servos may also include ball bearings which help tolerate loading at the servo's rotating interface.

Myoelectric Amplification Circuitry

While there are many sensory technologies which can be utilized to control the actuation of a prosthetic hand, an ultimate goal of human machine interfacing for prosthetic devices is to utilize control methods which are as similar as possible to the patient's original control of the lost upper extremity. One way off accomplishing this is by utilizing myoelectirc sensors. Myoelectric amplification circuitry is a technology which enables a device to detect the electrical impulses conducted by a patients nerves to muscles in their residual limb. These electrical impulses called action potentials are generated in the motor cortex of the brain and travel along the spinal cord and efferent nerves of the peripheral nervous system in order to cause contraction of specific muscle groups. Even in patients who have lost large portions of a muscle group or limb, many of these nerve networks remain intact with the residual muscle and these signals can be amplified through analog circuitry in order to use electrical signals from the brain as a direct control input for a prosthetic device. With prosthetic

devices these signals are normally detected using surface electrodes (Figure 11) which are capable of detecting the action potentials through the skin.

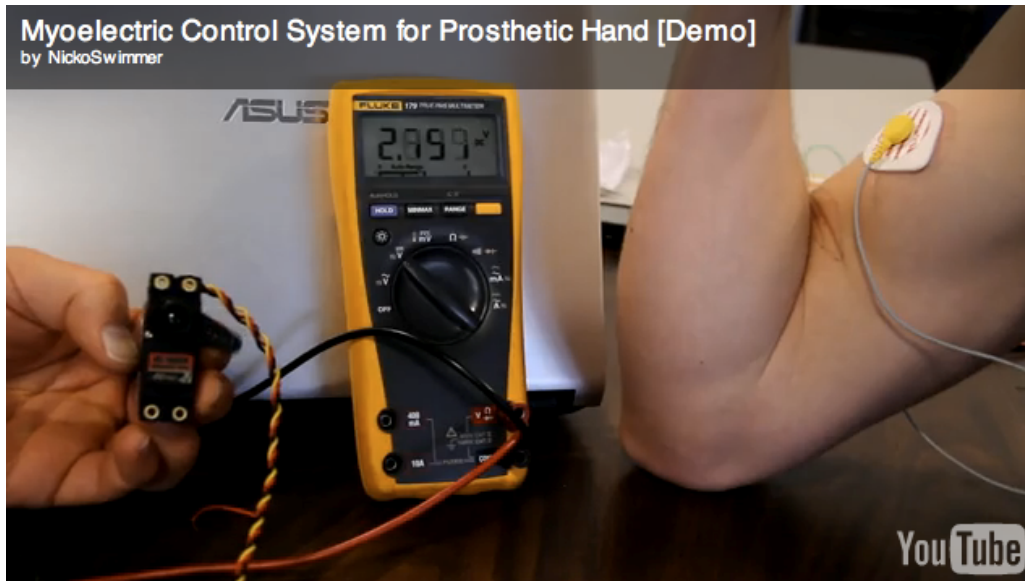


Figure 11 Myoelectric circuit showing a subjects muscle contraction being detected and used as a control input for a servo (Electrical components of the PolyGrasp Hand developed by Nickolas Butler, Matt Greibel, and Max Maloney)

Furthermore surface electromyography (sEMG) has researched since the Sixties to actuate dexterous prosthetic hand prototypes [15]. Surface electrodes are generally arranged with two electrodes positioned over the muscle group being measured and a reference electrode placed somewhere else on the subject to measure background noise associated with other muscle contractions and electrical activity in the body (Figure 12).

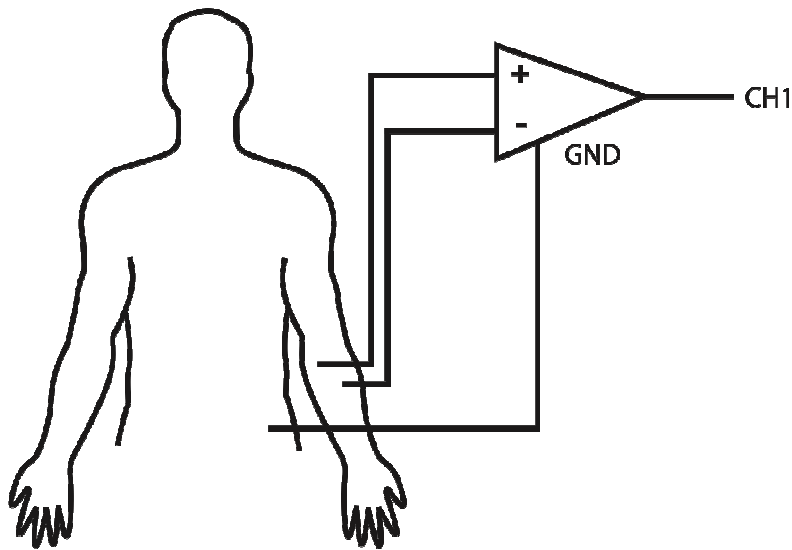


Figure 12 Diagram of the electrode placement for surface for surface electromyography (sEMG)

Due to high noise caused by adjacent electrical activity of tissues in the body and interference from common electronic devices, little information can be extracted from a raw EMG signal (Figure 13) Consequently several steps to condition the signal are performed including front end amplifier signal conditioning, artifact reduction, filtering methods, and analog-to-digital conversion of the sEMG signal [16].

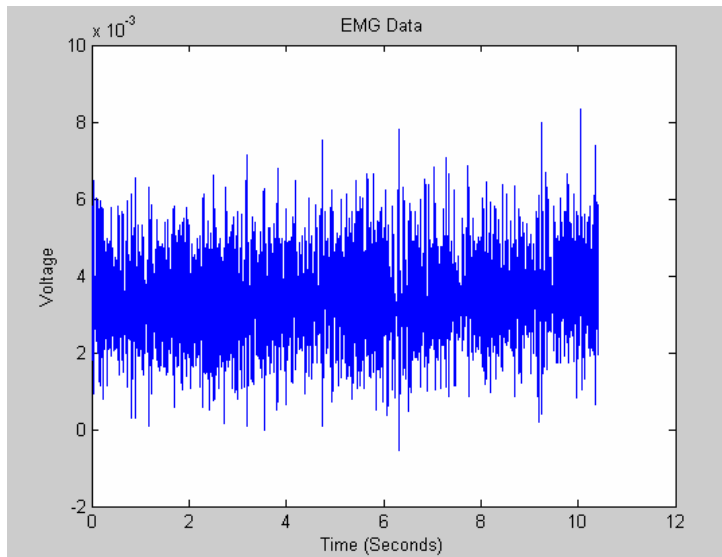


Figure 13Unfiltered EMG signal showing the high levels of noise caused by adjacent electrical activity in bodily tissues and interference from common electrical devices

The signal from each electrode is amplified and filtered to isolate the frequency range of action potentials. The measuring electrodes over the muscle group are then compared to the reference electrode using a differential amplifier which further removes background noise from a given signal. Through further digital processing, including power spectrum analysis and rectification of the filtered EMG signal, a more usable EMG waveform can be produced (Figure 14).

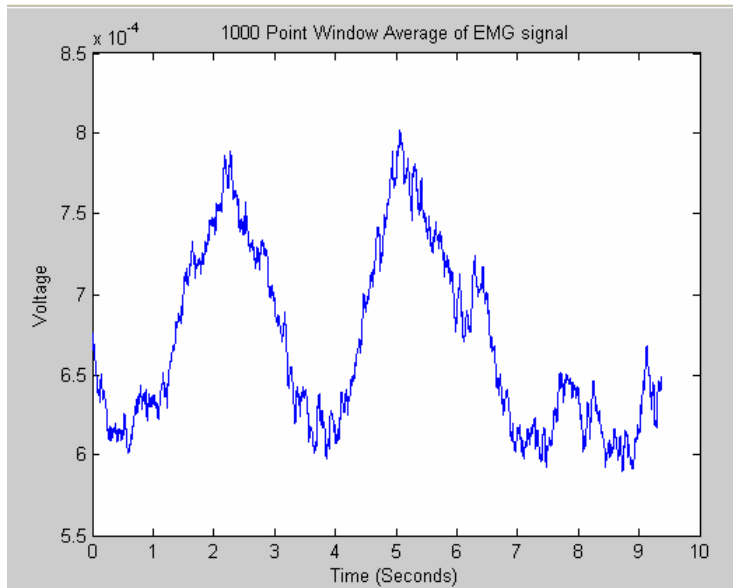


Figure 14 1000 point averaging windows of the full-wave rectified version of the EMG signal

With further digital input algorithms and signal smoothing this EMG signal can then be utilized as an input for prosthetic device.

Myoelectric Circuitry at Cal Poly San Luis Obispo

The development of myoelectric circuitry has been the focus of several projects at Cal Poly San Luis Obispo. One such project was the development of the Poly Grasp hand by Nickolas Butler, Matt Griebel, and Max Maloney through the Quality of Life Plus (QL+) program. The PolyGrasp hand featured a single channel myoelectric circuit which read electrical impulses in the forearm in order to actuate a large RC servo connected to a servo controller circuit (Figure 15).

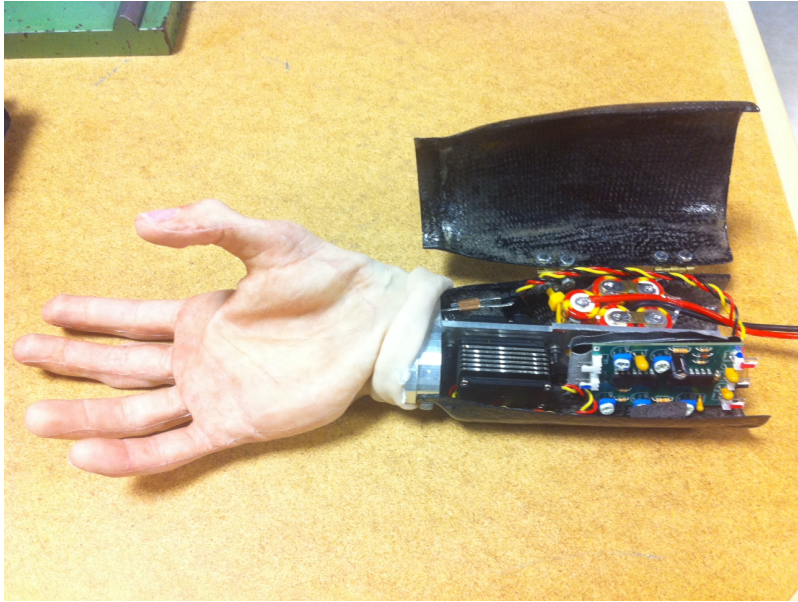


Figure 15 The myoelectric amplification circuit used in the PolyGrasp hand developed for the QL+ program (Project Poly Grasp: Nicolas Butler, Matt Griebel, and Max Maloney)

This myoelectric circuit was further developed as part of a thesis project in which the myoelectric circuit was further characterized and run in parallel with an identical circuit in order to produce two myoelectric output signals.

Prosthetic Hand Designs Through Cal Poly and Quality of Life Plus

Two upper extremity prosthetic devices have been developed at Cal Poly San Luis Obispo through the QL+ program. The PolyGrasp hand was designed to read myoelectric stimuli from the forearm in order to actuate a larger servo connected to an analog servo controller. This servo actuated two fingers corresponding to the “index” and “middle” fingers of the hand. The other fingers and thumb were non-functional. The entire assembly was fitted with a highly realistic silicone covering which closely resembled the skin tone and features of a normal hand.

Another upper extremity prosthetic developed through the QL+ program was designed to be used by an active duty Navy SEAL and was thus engineered to withstand harsh environmental and use conditions. The prosthetic was actuated by a hybrid cable system which could be manually actuated by the subject's residual palm and using modified version of the PolyGrasp hand's myoelectric circuit (Figure 16).

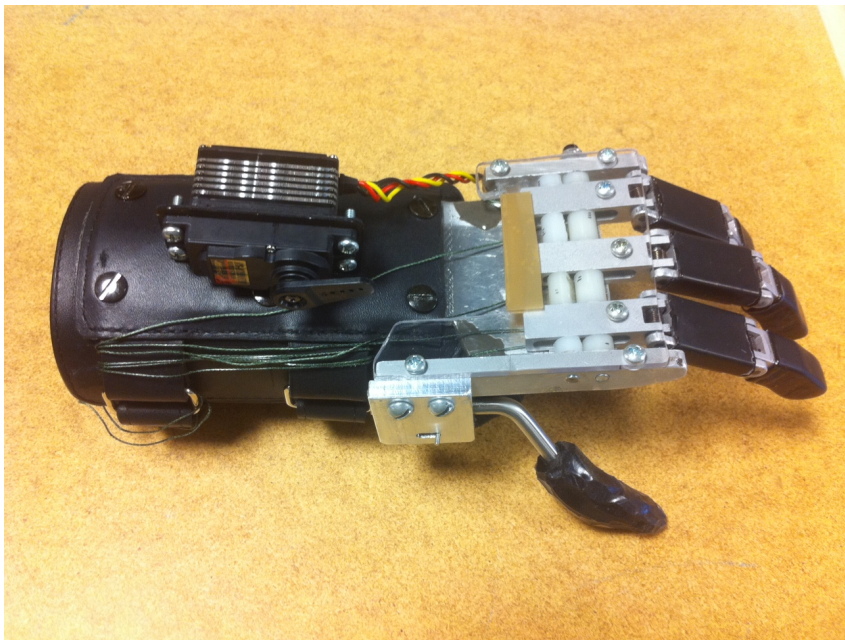


Figure 16 The prosthetic hand developed for a navy SEAL through the QL+ program

Flexion of the palm resulted in closure of the fingers through a cable actuation system which was anchored to the wrist. Additional features of the project included a locking multi-position thumb, a modular rail system for accessory attachment, and a modular finger attachment system which allowed for anthropomorphic positioning and easy replacement of fingers on the device.

3.0 DESIGN PROCESS

Goals

This project focused on the development of a prosthetic hand which utilizes an open source microcontroller and off the shelf components to provide the advanced functionality of available electronic prosthetics at a fraction of the cost. By minimizing the cost of this platform, a wider range of patients will be able to consider utilizing a more dexterous electric prosthetic hand in addition to their conventional mechanical prosthesis. The hand will utilize a variety of input sensors and monitor the load on its actuators in order to adjust performance in real time based on its current operational mode. A final prototype will be produced to showcase these features and augment the functions demonstrated by this projects proof of concept prototype.

Functional & Non-Functional Requirements

In order to meet the goals of this thesis project, the following list of functional and non-functional requirements was compiled. These specifications detail the primary areas of focus for this thesis which are intended to showcase the benefits of the development of an open source prosthetic hand platform from off the shelf components.

1. The hand will produce sufficient gripping force to be useful in light to medium lifting tasks.

2. The design will demonstrate adequate battery life to accommodate normal use.
3. Hand will be able to toggle between various operational modes through actions that can be performed by the prosthetic patient.
4. Design will utilize off the shelf electronics and open source microcontroller technology.
5. Mechanical design will be sufficiently durable such that components will not fail during normal use.
6. The electronics will be shielded from damage by a protective enclosure.
7. Cost shall not significantly exceed the \$500 budget awarded from the Hannah Forbes Senior Project Fund.

Desired Design Features

1. Fingers will adjust in real time to force sensed in the fingers.
2. Fingers will utilize a novel closure mechanism to improve on existing cable-actuated systems.
3. Design will be capable of adaptation to actual use by patients with minimal functional changes.
4. Design will have an anthropomorphic finger design, which may or may not be concealed per the patient's preference.

Design Assumptions

1. Final deliverable will be optimized for demonstration of functional capabilities.

- a. Mechanical components will be designed for manufacturability
 - b. Electrical circuitry will be configured in prototype form
- 2. Control inputs including pressure sensing pads, accelerometers, and myoelectric circuitry will be considered and incorporated into the final control system.

Project Timeline - Proposed and Actual

This timeline was created to allow for sufficient time for the design, prototyping, and testing. There are incremental deadlines for the completion of various parts of this projects major components with a projected defense date of June 2011.

Table I Timetable of thesis tasks with proposed and actual completion dates

Task	Projected Completion Date	Actual Completion Date
Develop proof of concept prototype	April 15 2010	April 15 2010
Complete documentation of prototype development	June 15 2010	June 15 2010
Develop stand alone control board	August 15 2010	August 15 2010
Complete documentation of rev1. Design and testing (methods & results)	August 31 2010	September 30 2010
Review completed rev1. thesis components with advisor	September 30 2010	September 30 2010
Design board for integration with prosthetic PCB	October 15 2010	October 15 2010
Test integrated prosthetic in demanding field use conditions	October 15 2010	December 30 2010
Complete documentation of rev2. Design and testing (methods & results)	October 30 2010	December 30 2010
Review completed rev2. thesis components with advisor	October 30 2010	April 25 2011
Complete thesis introduction	November 30 2010	April 1 2011
Complete thesis discussion	January 15 2011	April 15 2011
Complete thesis abstract	January 20 2011	April 15 2011
Review compiled thesis	January 31 2011	April 22 2011
Submit thesis for committee read	February 15 2011	May 23 2011
Develop Public thesis presentation	February 30 2011	June 3 2011
Defend thesis and discuss with advisors	March 4 2011	June 3 2011

Proof of Concept Prototype

The Rev. 1 prototype was initially constructed to demonstrate the capabilities of the open source Arduino microcontroller platform to coordinate the electrical systems of a prosthetic hand (Figure 17).



Figure 17 proof of concept prototype developed to demonstrate independent servos control and multiple operational modes using the arduino microcontroller

This prototype was designed as a rapid proof-of-concept from readily available electronic components (Table II).

Table II List of components that were used to construct the open source prosthetic hand's proof of concept prototype

Component	Quantity
Arduino Duemilanove	1
Hitec HS-81 servo	2
Force sensitive resistor	1
Small vibration motor	1
Momentary switch button	1
9V power supply	1

The electrical components were configured on a cardboard frame which was cut out in the shape a hand maintaining a neutral grasp. Separate cardboard cutouts were also made for the index finger and the remaining middle, ring, and little fingers of the hand on a single cut out. These finger group cut-outs were mounted to servos which allowed them to independently open and close using a FSR mounted on the side of the prototype in order to showcase the multiple operational modes of the prototype. The Rev.1 prototype could be toggled

between 4 operational modes using a button on the underside of the prototype (Table III).

Table III Operational modes designed into the firmware of the proof of concept prototype hand

Mode #	Function
1	Input signal actuates “index” and “middle-ring-little” finger groups with force feedback vibration
2	Clinch independent of input signal
3	Clinch “middle-ring-little” finger group and “index” finger held at ready trigger position
4	Clinch “middle-ring-little” finger group and “index” finger actuates with input signal

Depending on the operational mode, the proof of concept prototype would read the change in resistance of a force sensitive resistor (FSR) and result in a specific combination of servo actuation, vibration feedback, and LED activation to indicate its mode state. The proof-of-concept prototype demonstrated that these actions could be coordinated effectively by the Arduino microcontroller providing the proof-of-concept to justify further development. The initial code uploaded to the microcontroller (Appendix B) was divided into sub modules and validated subsequent sections.

Servos

A servo control module can allow for the servos connected to the Arduino microcontroller to move between predetermined rotational positions. This module provides a foundation for future code which will actuate the fingers of the prosthetic hand to specific positions based on hand’s other sensory inputs and current operational mode. The servo code modules: ServoRock and ServoRock2

(Appendix A) allowed for either 1 or 2 servos to be connected to the pulse width modulation capable digital pins of the Arduino Mini board (Figure 18).

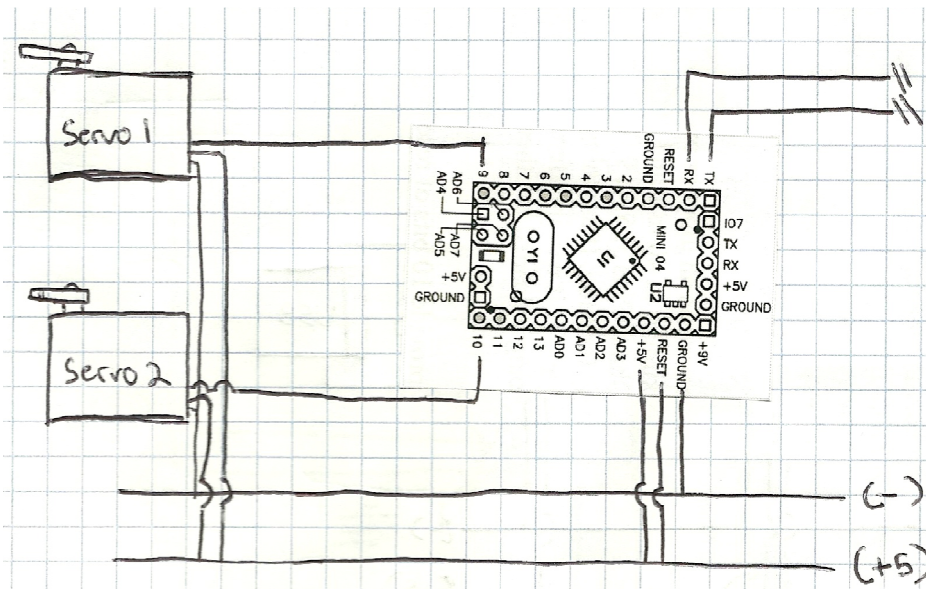


Figure 18 Schematic of servo integration with the Arduino “Stamp” Mini

The servos were each wired to a pulse width modulated (PWM) digital pin, ground, and +5V. These servos would oscillate at a given rate from one specified extreme of the servo’s 60 degree range to another based on the control signal sent from the Arduino over the PWM pin.

Force Sensitive Resistor

A force sensitive resistor module allows for the pressure exerted on an FSR to be read by the Arduino microcontroller. The Microcontroller will use this input as a control signal to actuate the fingers of the prosthetic hand and also to measure the pressure sensed at the finger tips and relay force feedback to the user through vibrating motors. The force sensitive resistor module: FSR

In this configuration the Arduino Mini would print over the serial connection a number between 0 and 130 corresponding to either a hard or soft press on the FSR.

A module which detects a button press event is a useful function that allows the Arduino microcontroller to correctly determine if a button has been pressed and released. Because a button will read as High when depressed and Low when released, it is important that the module to detect a button press recognizes both of these events without accidentally reading a single button press event as multiple presses by simply seeing if the button is in the High state with

each cycle of the program. The button case change framework:

ButtonCaseChangeFramework (Appendix A) detected the pressing of a momentary switch button which resulted in either a digital open or closed signal. The button was wired at one side to one of the Arduino Mini's digital pins, and wired in series from +5V to the button, to a 10K ohm pull down resistor then to ground (Figure 20).

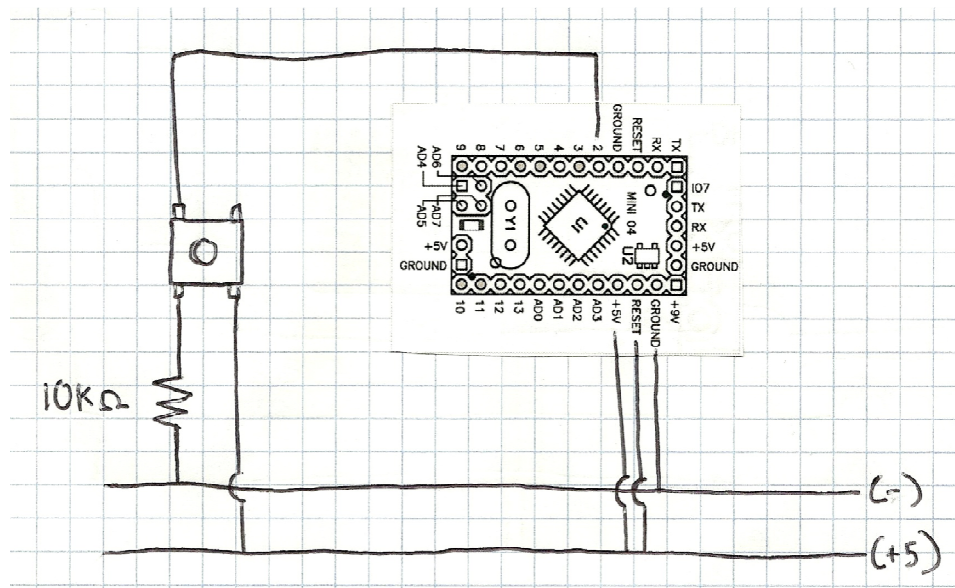


Figure 20 Schematic of momentary switch integration with the Arduino “Stamp” Mini

In this configuration the Arduino Mini would print over the serial connection either a 0, 1, 2, or 3 corresponding to one of the four operational modes.

Current Sensing Module

The current sensing module: CurrentSensingModule (Appendix A) was designed to measure the change in current draw by a servo as it is placed under increasing load. The Arduino Mini was connected to a current sensing module

which would vary its output voltage depending on the current drawn by the servo (Figure 21).

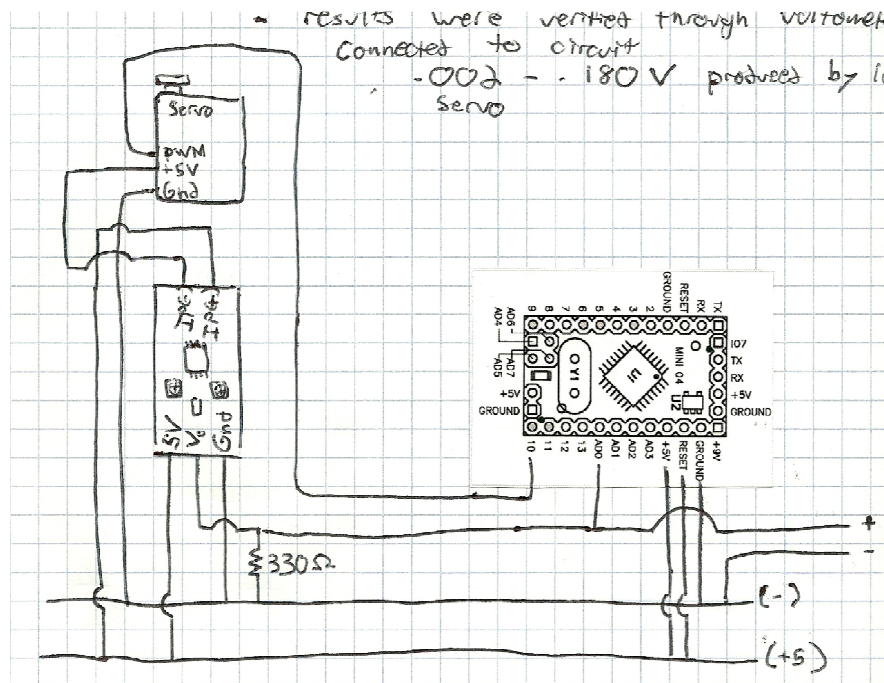


Figure 21 Schematic of current sensor integration with the Arduino “Stamp” Mini

When connected to a voltmeter, the output voltage corresponding to current draw showed that when unloaded the servo would draw a steady lower current than when loaded with either a light or heavy force. This contradicted the assumption that gradually increased load would result in a corresponding increase in current draw. Also, this minor increase in current draw between unloaded and loaded servo states was not significant enough to observe on the microcontroller without amplification of the input voltage.

Two Axis Accelerometer

The Two Access Accelerometer: Accelerometer Sensor (Appendix A) was designed to read the pulse widths generated by a two access accelerometer

which would correspond to the angle of tilt in each axis. The pulse width would then be converted to a value between -1000 and 1000 then printed over the serial connection. Then each axis of the accelerometer was connected to a digital pin on the arduino in addition to ground and +5V (Figure 22).

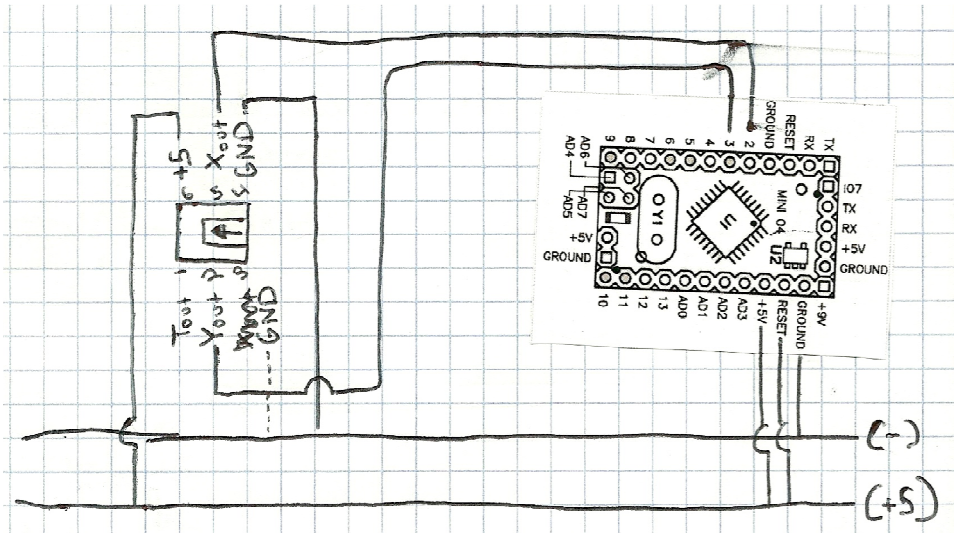


Figure 22 Schematic of two axis accelerometer integration with the Arduino “Stamp” Mini

Myoelectric Input Integration

Initial tests were conducted to determine if the FSR circuit used to actuate the fingers of the prosthetic hand could be easily replaced using a myoelectric circuit. This was achieved using the single channel myoelectric circuit developed by Nickolas Butler [16]. This circuit was connected to the analog input and also to the common ground of the Arduino microcontroller. Initial tests of this integrated system showed promising results with the myoelectric circuit successfully being able to fully open and close the hand when reading muscle contraction of the flexor carpi ulnaris. However opening and closing of the prosthetic hand was actually opposite to the physical action of the human subject

resulting in contraction of the subject causing the hand to open and vice versa. This result was not surprising as the myoelectric circuit is an amplification of muscle action potentials and thus a positive amplitude reading while the FSR circuit measures the increased resistance of the FSR resulting in a decreasing amplitude signal. Ultimately if integrated into final product, this defect could easily be addressed programmatically by inverting the input parameter from the myoelectric circuit. Furthermore smoothing of the analog signal from the myoelectric circuit could also be investigated in order to reduce tremor of the fingers in intermediate (not completely closed or open) positions.

Open Hand 1.1 Code (See OpenHand I.0)

The first algorithm used to coordinate the motions of the hand utilized the previously characterized button case change framework algorithm to toggle between four operational modes in sequence then returning to the initial mode (Appendix B). The first mode used the FSR input to actuate the “index” and “middle-ring-little” finger groups with force feedback vibration. The second mode caused the fingers to enter into a closed clinch position independent of input signal. The third algorithm clinched the “middle-ring-little” finger groups and the “index” finger was positioned at an open trigger position and would not respond to FSR input. The forth mode again clinched the “middle-ring-little” finger groups and “index” finger actuated with the FSR input signal. This algorithm is displayed in the following block diagram (Figure 23).

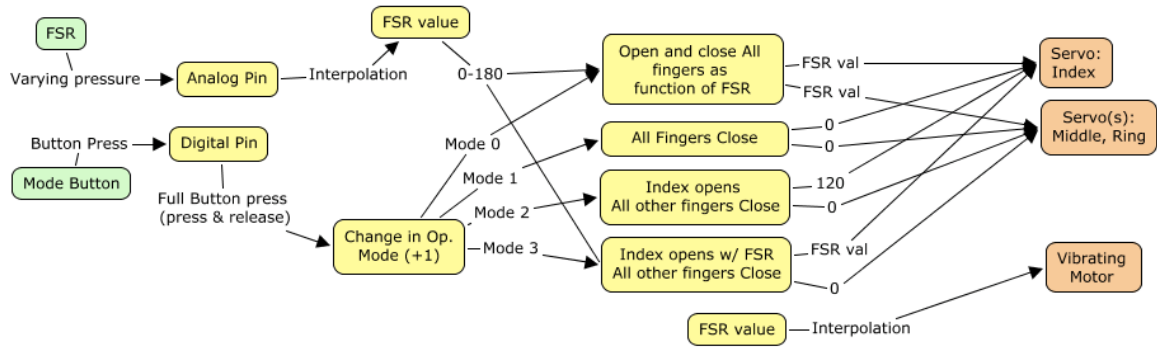


Figure 23 A block diagram of the Open Hand 1.1 code with the effects of the input electronic components (green) on the Arduino’s programmed algorithms (yellow) and their resulting effects on the hand’s outputs (red). The Open Hand 1.1 code features 4 operational modes which affect the functions of the hand’s fingers and are toggled between using a mode button

Open Hand 1.2 Code

The second algorithm used to coordinate the motions and feedback of the hand utilized an input switch located in the thumb which depending on its open or closed state, toggled the hand between two operational modes (Appendix B). The first mode, toggled by opening the switch at the thumb allowed all the fingers to be actuated equally as a function of the pressure read by the FSR. In the second mode, the thumb is moved in line with the “index” finger group to close the thumb position circuit. This caused the “middle” and “ring” finger groups to enter a closed clinch position while the “index” finger group actuated as a function of the pressure read by the FSR. In both cases, the algorithm also took readings from pressure sensors in each finger, summed the total pressure read, and signaled a vibrating motor to vibrate with corresponding intensity. This algorithm is displayed in the following block diagram (Figure 24).

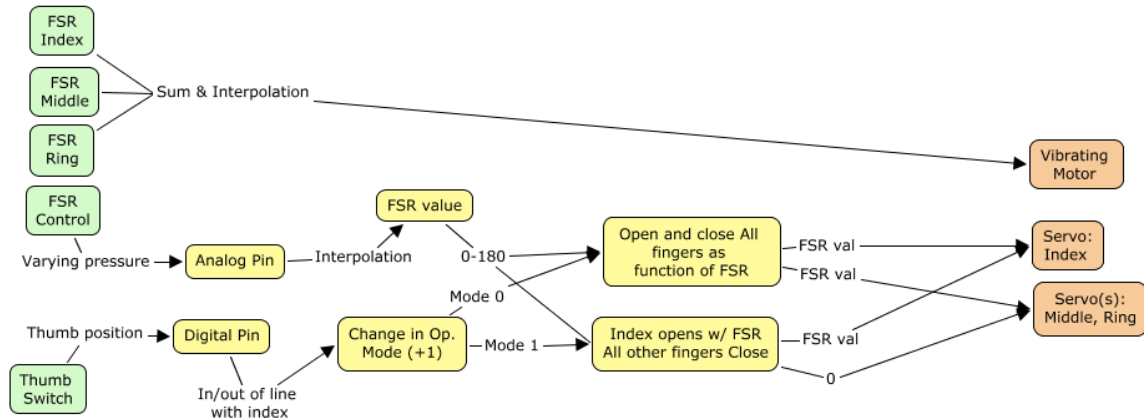


Figure 24 A block diagram of the Open Hand 1.2 code with the effects of the input electronic components (green) on the Arduino's programmed algorithms (yellow) and their resulting effects on the hand's outputs (red). The Open Hand 1.2 code features 2 operational modes which affect the functions of the hand's fingers and are toggled between using a switch which detects when the thumb is in line with the index finger

Rev. 1 Mechanical Design

A primary challenge of developing the open source prosthetic hand platform was the designing a hand which could be manufactured using available equipment at Cal Poly San Luis Obispo while optimizing grip strength and anthropometrics with respect to the patient's native anatomy. With this in mind, the purpose of the Rev.1 mechanical design was to showcase the desired form factor for the attachment of the fingers to the palm and the organization of the servos which actuate the fingers. The fingers utilized a standardized design which varied only in overall length allowing for the same parts to be utilized for different fingers while producing an anthropomorphic grasp. This design also established a preliminary layout for the three servos used to actuate the fingers themselves. One servo would actuate the "pinky" and "ring" fingers, the second would actuate the "middle" finger, the third would actuate the "index" finger, and lastly the thumb would be manually adjusted by the user (Figure 25).

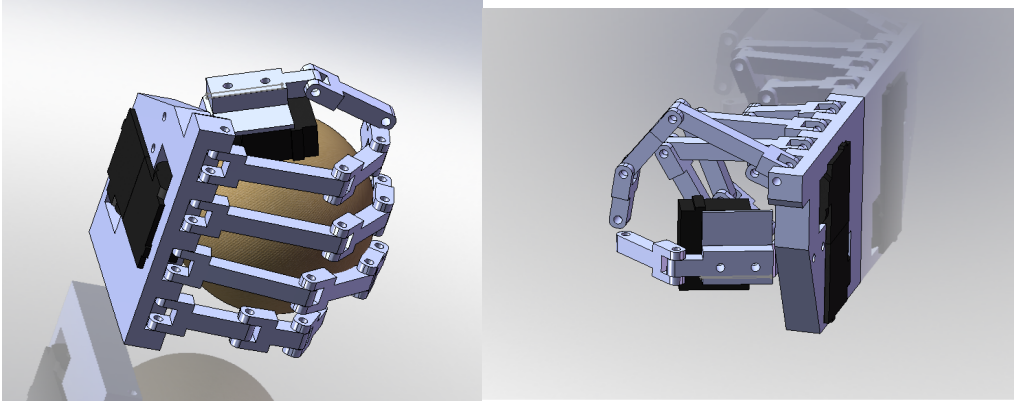


Figure 25 Rev. 1 mechanical design showing the arrangement of three servos used to power a five-fingered version of the prosthetic hand

Some deficiencies to this design included the lack of routing pathways for the cables/ rods used to actuate the fingers. This design also raised the possibility of encountering some grip weakness toward the open extreme of the grasp due to mechanically challenging orientation of the fingers in this position. Further consideration would also need to be made in regards to the location and attachment of the electrical components and sensors to the fingers and palm of the hand.

Rev. 2 Mechanical Design

The Rev.2 mechanical design focused on addressing some the concerns with utilizing cable driven finger actuation system. This design demonstrated using a system of levers to actuate the distal digits of the finger based on the orientation of the most proximal digit relative to the palm (Figure 26).

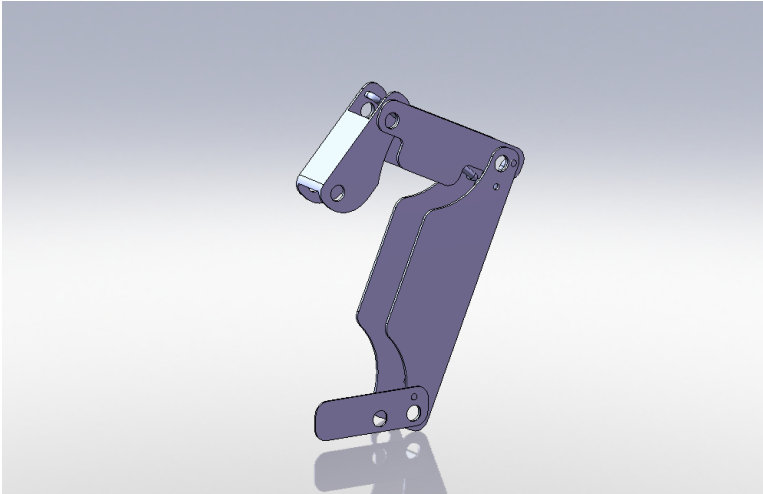


Figure 26 A finger design constructed from bent sheet metal with internal lever arms which cause bending of the distal portions of the finger

This design could be manufactured from sheet metal bent into the U-shaped components which make up the finger construction. The “Bend Rods” could then be attached at locations along the U-shaped finger structures to control bending of the finger during actuation (Figure 27).

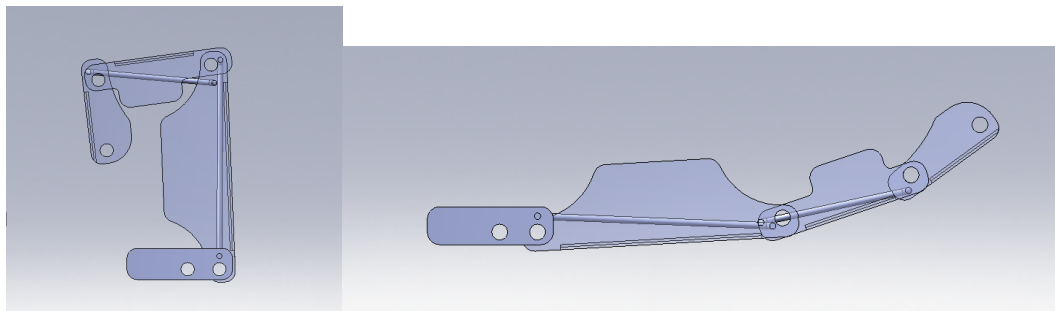


Figure 27 Auto-grip motion of the fingers caused by internal lever arms

Although this design created a possible alternative to a cable actuated finger system, there were several manufacturing challenges to this design. First, the design would require access to a CNC laser cutter capable of cutting sheet metal. Second, the design did not include the fastening hardware for the finger joints. Lastly the design did not detail the attachment and fixation method for the

“bending rods” which left to question how this aspect of the design would be manufactured and made it difficult to predict the strength of these features.

Rev. 3 Mechanical Design

The Rev.3 mechanical design also focused primarily on finger design although it builds on the Rev.2 mechanical design by proposing the hardware used to assemble the joints of the device in addition to the off the shelf materials used to construct the digits as well. Furthermore this design eliminated bending at the most distal finger joint due to the added complexity and minimal benefit. This design can be broken into two versions: the Rev. 3 mechanical concept, and the Rev. 3 functional prototype.

The Rev. 3 mechanical concept featured an anthropomorphic design with finger digits which could be machined or injection molded depending on the desired complexity of finished features. This design also featured off the shelf joint hardware and replaced the “bending rods” of the Rev. 2 design with aluminum bar stock which actually functions as a structural member in addition to controlling bending of the finger during actuation.

The Rev. 3 functional prototype (Figure 28) was functionally identical to the mechanical concept but instead replaced the middle digit with delrin bars that featured an identical hole pattern. This change simplified the overall design while maintaining the “auto-curling” function of the fingers as they are actuated. The distal digit was also simplified to be laser cut from .25” delrin. This design allows for quick prototyping of the design in order for the mechanical properties of the design to be evaluated.

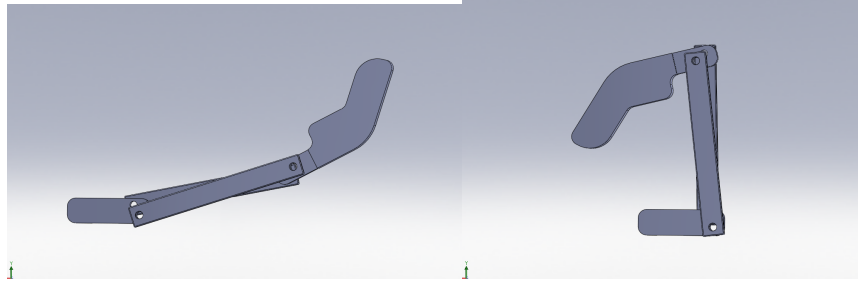


Figure 28 A simplified finger bending mechanism which utilizes juxtaposed levers that also function as structural elements. These levers allow for the distal tip of the finger to curl inward as the two levers are flexed toward the palm

An additional benefit of this version of the Rev 3 design is that the design utilizes less sophisticated manufacturing processes and fewer steps to produce which would reduce the cost of producing fingers with this design.

Rev 3 Mechanical Design (Thumb)

This design of the thumb incorporated a set of mechanical stops in addition to a small switch which allowed the thumb to be set in discrete positions and provide a basic signal to the microcontroller indicating its orientation. (Figure 29)

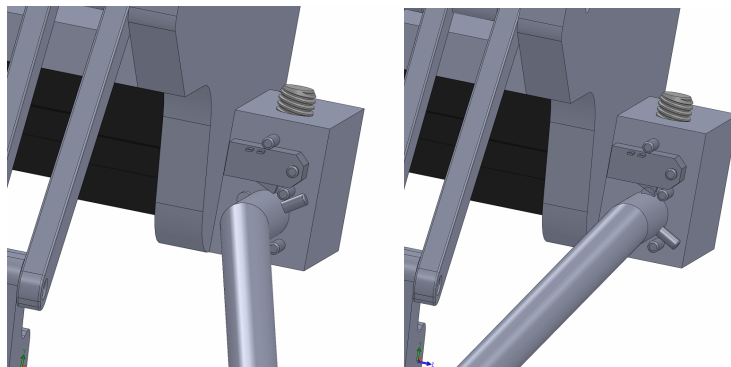


Figure 29 A switch located in the thumb housing which signals the position of the thumb to the hand's microcontroller

The thumb is able to remain in discrete open and closed positions due to the mechanical stops created by dowel pins in the thumb rod and base. Slippage

between these positions was reduced by installing a ball plunger in the thumb base which held the thumb in the open and closed positions. The switch would then generate either an open or low signal when the thumb was in the extended position and would read a closed or high signal when the thumb was moved in line with the index finger. This High, or low signal produced by the thumb switch could then be read by one of the Arduino's digital pins and toggle the hand's microcontroller into a different operational mode based on this signal.

4.0 FINAL DESIGN

Final Prototype

The final prototype for the open source prosthetic hand built upon the design and programming considerations gained through the initial design process. The hand consists of a palm section which contains the fingers, servos, and input sensors and a wrist portion which contains the batteries, microcontroller, and sensory outputs (Figure 30).



Figure 30 The final prototype of the open source prosthetic hand platform featuring three independently actuated fingers and a thumb which can be utilized to toggle between grasping and pinching modes of operation

The final prototype utilized the simplified design of the rev 3 fingers. The digits were laser cut from .125 in and .25 in black delrin stock and assembled

with dowel pins at the proximal joint and small shoulder screws at the distal digit. An anthropometric grip was achieved by varying the length of the most proximal digit. The hand featured three fingers with the middle proximal digit .25 in longer than outside fingers (Figure 31).

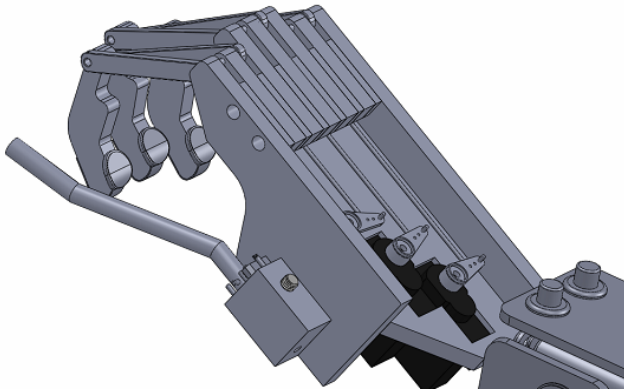


Figure 31 The palm portion of the open source prosthetic hand which utilizes a modular finger attachment mechanism in order to create an anthropometric grip

The fingers were designed to utilize mostly identical components to facilitate easy replacement. These components were assembled using 10-32 threaded bolts which hold the fingers in a horizontally stacked position between two .25in panels which make up the thumb and little finger sides of palm.

A force sensitive resistor was attached to the distal portions of each finger allowing for the forces transferred to this part of the finger to be read by the hand's microcontroller (Figure 32).

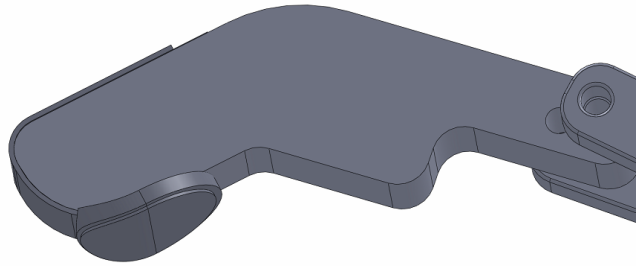


Figure 32 The positioning of a force sensitive resistor on the distal portion of the fingers in order to measure grip pressure at the fingertip

This sensor was attached to the finger tips using hot glue at the sensor finger interface. The sensor was also coated with hot glue in a rippled pattern to improve grip and force detection by sensor.

The base portion of the palm housing featured a laser cut slotted geometry which allowed the servos to be closely arranged in a space efficient configuration. The servos were connected by ridged supports to the base of the fingers (Figure 33).

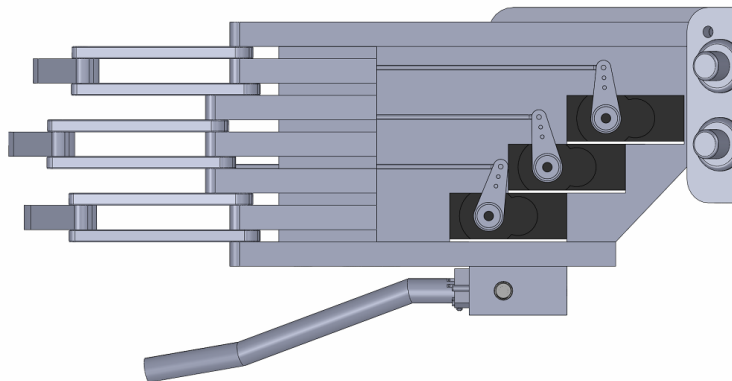


Figure 33 The push pull rods used to open and close the hands fingers

This allowed the fingers to be both opened and closed by the servos without the use of springs to restore the fingers to a neutral position.

The last major component of the palm portion was the discrete positional thumb which was not altered from its design described in Rev 3 of the design process.

The wrist portion of the prosthetic hand houses the majority of the hand's electronic components. The wrist is attached to the palm via a .125in aluminum plate with a 30° incline which recreates the approximate anthropometrics of the wrist at a neutral position (Figure 34).

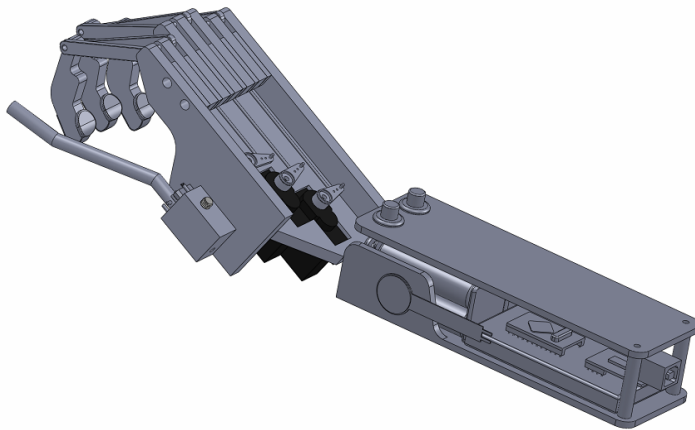


Figure 34 The final prosthetic hand prototype showing the 30 degree incline at the wrist to match the neutral positioning of a patients native hand

The base portion of the wrist secures the batteries, electronics, prototype board, and the wrist cover plate. The wrist base also features an inverted tab which provides a sturdy base for a force sensitive resistor used as an input to demo the operation of the hand. The wrist cover plate features two switches which connect a 6V, 4AA battery power supply to the servos and a 9V power supply to microcontroller and sensors. Two power supplies were used so that heavy use of

the servos would not result in under powering and shutdown of the microcontroller, also known as brownout. A vibrating motor was attached to the wrist cover plate which allowed for forces sensed by the force sensitive resistors in the fingers to be output in the form of force feedback (Figure 35).

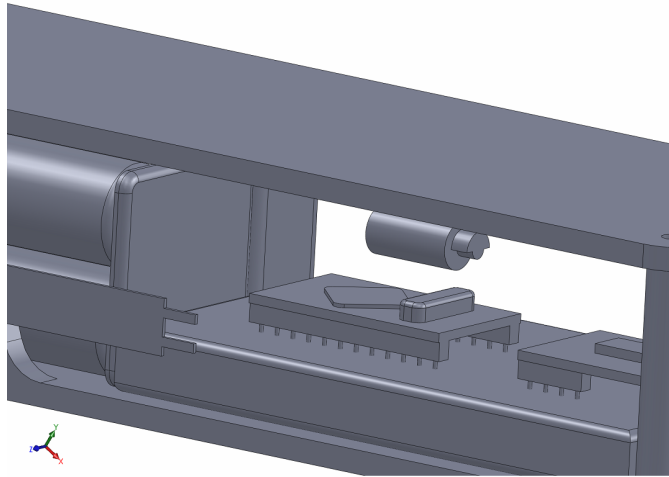


Figure 35 A vibrating motor located in the wrist portion of the prosthetic hand used to provide force feed back to the user

The code used to coordinate the actions of the final prototype, Open hand 1.3, was utilized in the final design prototype. This code followed the same algorithm as Open Hand 1.2 (Figure 24) and allowed for inputs including the force sensitive resistors, and thumb position sensor to be used to actuate servos, in either a grasping motion operation (Figure 36 A & B), or a pinching motion (Figure 36 C & D).

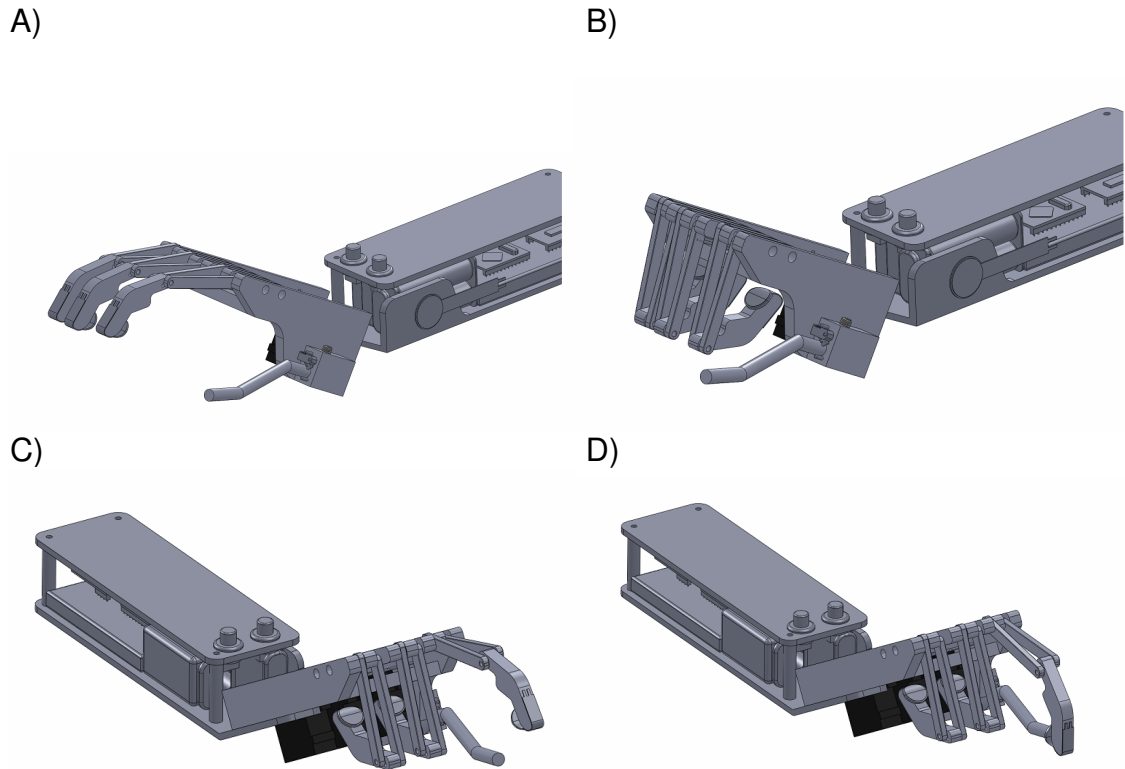


Figure 36 The open and closed grasp position (A&B) and the open and closed pinch position (C&D) of the final prototype hand

A schematic showing the wiring of the Arduino microcontroller and the final prototype's electrical components is shown in Figure 37.

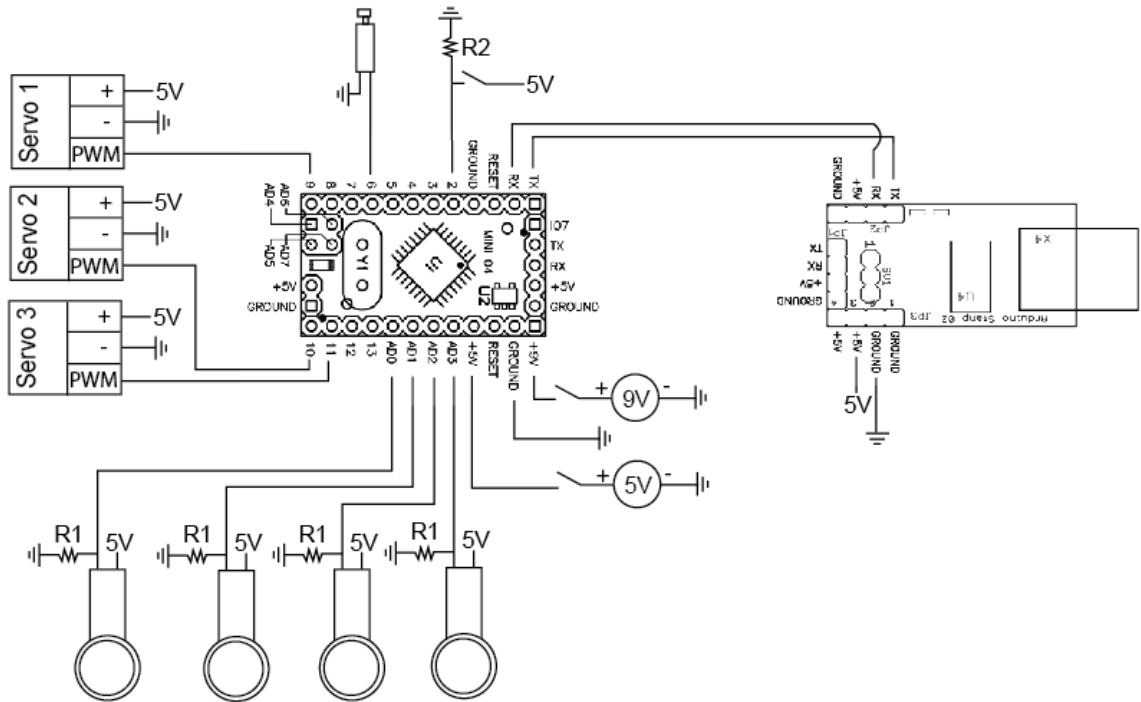


Figure 37 A schematic of the electronics utilized in the final prototype. The hand features 4 FSRs which are used as inputs for the hand's 3 servos causing finger actuation and force feedback from a vibrating motor. The optional connections for the microcontroller's USB interface is also shown

Three of the analog inputs of the microcontroller will be allotted to the force sensitive resistors in each of the three retracting fingers. A fourth FSR will be used as the primary control input for the purpose of demonstrating the hands functionality. Three PWM capable digital channels will be used by the servos in addition to one other PWM channel for the vibrating motor. Lastly a non-PWM digital channel will be used to read whether the thumb is in the open or closed position. This configuration will allow for the thumb position to determine if the input received from the control FSR will be used to drive all three servos in the grasp mode or if the middle and ring fingers will be closed and only the index finger will be controlled in the pinch mode. The USB connection board will also be included in the final prototype in order to run diagnostic tests on the hand and

update the microcontrollers firmware. Switches were placed in series with the 9V power supply to the microcontroller, position switch and FSRs as well as between the 6V, 4AA power supply and the three finger servos.

Component List and Prices

The following (Table IV) is a list of the components, vendors, and costs for all materials associated with the development of the open source prosthetic hand platform. Larger vendors were used in order for components to be reordered easily if necessary.

Table IV Bill of materials detailing all purchase orders associated with the development of the open source prosthetic hand platform including materials not necessarily included in the final prototype

Description	Part #	Vendor	Price	order # / reciept	
Arduino mini Microcontroller	DEV-08164	Sparkfun.com	\$33.95	SF 227534	
Arduino Serial USB Board	DEV-08165	Sparkfun.com	\$20.95		
ACS712 Low Current Sensor Breakout	SEN-08883	Sparkfun.com	\$14.95		
Breadboard (x2)	PRT-09567	Sparkfun.com	\$11.90		
Force Sensitive Resistor – Small	SEN-09673	Sparkfun.com	\$5.95		
Mini Push Button Switch (x2)	COM-08720	Sparkfun.com	\$1.90		
Resistor kit	COM-09258	Sparkfun.com	\$10.95		
			\$9.46		
SF 227534 Shipping		Sparkfun.com		Sub total:	\$110.01
Hitec HS-82MG Servo (x3)	HRC32125S	Rcsuperstore.com	\$65.97	RCS 10960	
RCS 10960 Shipping			\$6.95		
		Rcsuperstore.co		Subtotal	\$72.92
Memsic 2125 Dual-axis Accelerometer	MKPX7	Makershed.com	\$29.99	MS 60340	
MS 60340 Shipping		Makershed.com	\$7.20		
MS 60340 Tax			\$2.77		
		Makershed.com		Subtotal	\$39.96
Force Sensitive Resistor .5" (x5)	SEN-09375	Sparkfun.com	\$34.75	SF 265722	
			\$4.41		
SF 265722 Shipping		Sparkfun.com		Subtotal	\$39.16
40 Pin IC socket	2751996	Radioshack	\$0.99	774293	
EXP300 PC bard	2760170	Radioshack	\$2.99		
EXP300 PC bard	2760170	Radioshack	\$2.99		
PCB Term 2p 5mm	2761388	Radioshack	\$1.99		
Snap Bat Hldr 4AA	2700383	Radioshack	\$1.99		
4AA Battery hldr	2700391	Radioshack	\$1.99		
9V & AA battery packs			\$10.00		
	Special	Radioshack			
tax		Radioshack	\$2.01	Subtotal	\$24.95
324 Quad op amp	2761711	Radioshack	\$1.69	567053	
100K micro pot	2710284	Radioshack	\$1.49		
100K micro pot	2710284	Radioshack	\$1.49		
1K micro pot	2710280	Radioshack	\$1.49		
1K micro pot	2710280	Radioshack	\$1.49		
1K micro pot	2710280	Radioshack	\$1.49		
100K Micro pot	2710284	Radioshack	\$1.49		
90' #22 Solid UL	2781221	Radioshack	\$7.99		
tax		Radioshack	\$1.82	Subtotal	\$20.44
Shoulder Screw (x8)	99154A301	McMaster Carr	\$31.44	0922SGARRETT	
Arch AL 1/8"	4490T14	McMaster Carr	\$2.86		
Black Delrin 1/8"	8575K113	McMaster Carr	\$14.38		
Black Delrin 1/4"	8575K115	McMaster Carr	\$25.27		
18-8 dowel Pins 1/16"	90145A417	McMaster Carr	\$8.08		
tax		McMaster Carr	\$6.77		
Shipping		McMaster Carr	\$16.91	Subtotal	\$105.71
Total			\$413.15		

The final cost of components was \$413.15 which was below the \$500.00 budget allotted through the Hannah Forbes Project Fund. The cost of components utilized for only the final prototype totaled \$311.21 and is detailed in Table V.

Table V Bill of materials detailing only materials and components included in the final prototype

	Description	Part #	Vendor	Price
Raw Materials and hardware	Shoulder Screw (x8)	99154A301	McMaster Carr	\$31.44
	Arch AL 1/8"	4490T14	McMaster Carr	\$2.86
	Black Delrin 1/8"	8575K113	McMaster Carr	\$14.38
	Black Delrin 1/4"	8575K115	McMaster Carr	\$25.27
	Black Delrin 1/2"	8662K42	McMaster Carr	\$8.89
	18-8 dowel Pins 1/16"	90145A417	McMaster Carr	\$8.08
	1/4" AL rod stock	6750K131	McMaster Carr	\$3.41
	Thin Rod stock (paper clips)		Staples	\$1.00
	Misc fasteners		Home depot	\$10.00
Electronics	Arduino mini Microcontroller	DEV-08164	Sparkfun.com	\$33.95
	Arduino Serial USB Board	DEV-08165	Sparkfun.com	\$20.95
	Breadboard	PRT-09567	Sparkfun.com	\$5.95
	Force Sensitive Resistor .5" (x5)	SEN-09375	Sparkfun.com	\$39.16
				\$1.99
	Relay switches (x2)		RadioShack	
	Hitec HS-82MG Servo (x3)	HRC32125S	Rcsuperstore.com	\$72.92
	Various resistors		RadioShack	\$5.00
	Wire		RadioShack	\$7.00
	4AA battery Holder		RadioShack	\$1.99
	9V battery Holder		RadioShack	\$1.99
	9V & AA Batteries		RadioShack	\$10.00
	Vibrating motor		RadioShack	\$2.99
	Wired detect switch		RadioShack	\$1.99
Total				\$311.21

5.0 TESTING AND RESULTS

After completing the final prototype of the open source prosthetic hand platform, it was necessary to characterize the hand's performance through a variety of functional tests. Properties such as the hand's power consumption, grip capacity, effectiveness in completing real-world tasks, and integration with advanced control inputs demonstrated the strengths of current platform in addition to areas for further development of the system.

Power Consumption Assessment

1. Purpose

The purpose of this lab is to quantify the performance of the prosthetic hand design during continuous use. By subjecting the hand to uniform sustained contractions of the fingers, the prolonged real world use characteristics of the hand can be better understood.

2. Description

The amount of energy consumed by the prosthetic hand will be characterized using state of charge (SOC) measurements take during a programmed power draining macro initiated on the prosthetic hand. Once started the macro will cause all fingers of the hand to contract and hold for approximately 5 seconds then release for 5 seconds and repeat this cycle for the duration of the test. A hand dynamometer will be placed in the palm of the hand to measure the decay of force production over time. Data from the dynamometer will be connected to an analog to digital signal converting device and plotted using the LabChart.

LabChart is a data acquisition program that can record and analyze signal from a variety of sensors during an experiment and present this data for later visualization or post processing. In addition to the hand dynamometer, a voltmeter will be used to measure potential across the servo battery in order to determine the state of charge at two minute intervals throughout the test.

3. Materials and Equipment

1. Open Hand Prosthetic Hand prototype with firmware version 1.3.1 (Appendix B)
2. Dynamometer adapter bar
3. AD Instruments Powerlab 26T
4. AD Instruments MLT0031D Hand Dynamometer
5. Multimeter
6. USB cable
7. Laptop with LabChart

4. Protocol

- 4.1. Connect the hand dynamometer to the Powerlab A to D converter and connect the converter to the analysis laptop via USB (Figure 38).

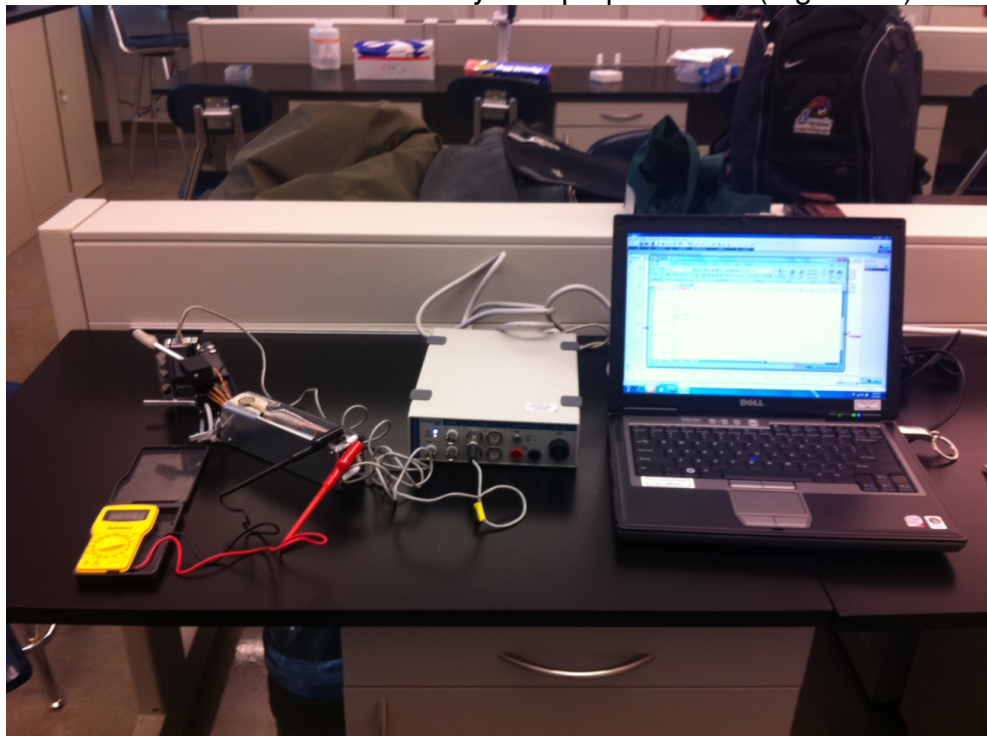


Figure 38 The setup used to measure battery drain using a multimeter and grip force using a hand dynamometer during the continuous use test

- 4.2. Power on all equipment and launch LabChart.
- 4.3. In LabChart configure channel 1 to read the mV input from the hand dynamometer.
- 4.4. Attach the dynamometer adapter bar to the palm of the prosthetic hand using zip ties to ensure consistent grasping of the dynamometer.
- 4.5. Place the dynamometer upright in the palm of the hand and align the long end of the force transducer against the adapter bar (Figure 39).



Figure 39 Attachment of an adapter bar to the hand using zip ties in order to align the force transducer in line with the grasp of the fingers

- 4.6. Connect the multimeter across the prosthetic's 6V servo power supply using the leads at the rear of the prosthetic and configure the multimeter to read less than 10V (Figure 40).

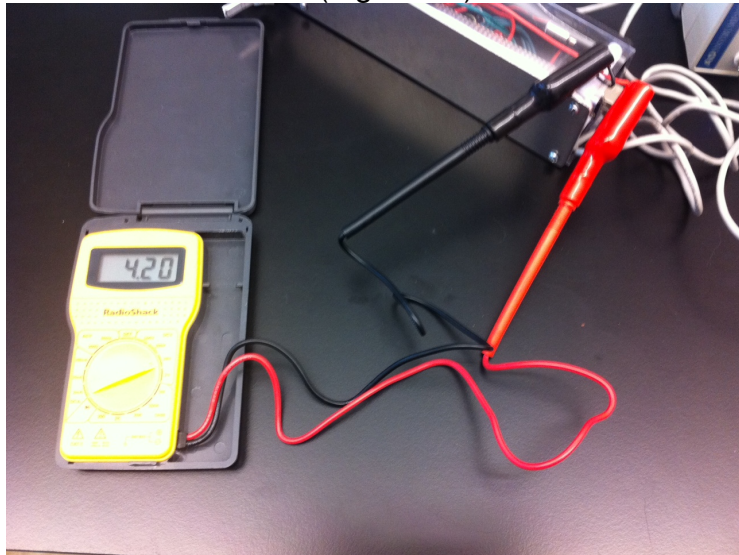


Figure 40 Connection of the multimeter to break out leads used to measure the potential across the servo power supply

- 4.7. Initiate the power draining subroutine.
- 4.8. Start the LabChart recording of the hand dynamometer force input.

- 4.9. Measure and record the potential across the prosthetic's 9V power supply at 2 minute intervals.
- 4.10. Monitor the hand during the power draining subroutine and record any observations.

5. Results

Following a more rapid decline in the first 20 minutes of the continuous use test the servo power supply showed a steady decline in potential of approximately 7.5 mV per minute for the duration of the test (Figure 41).

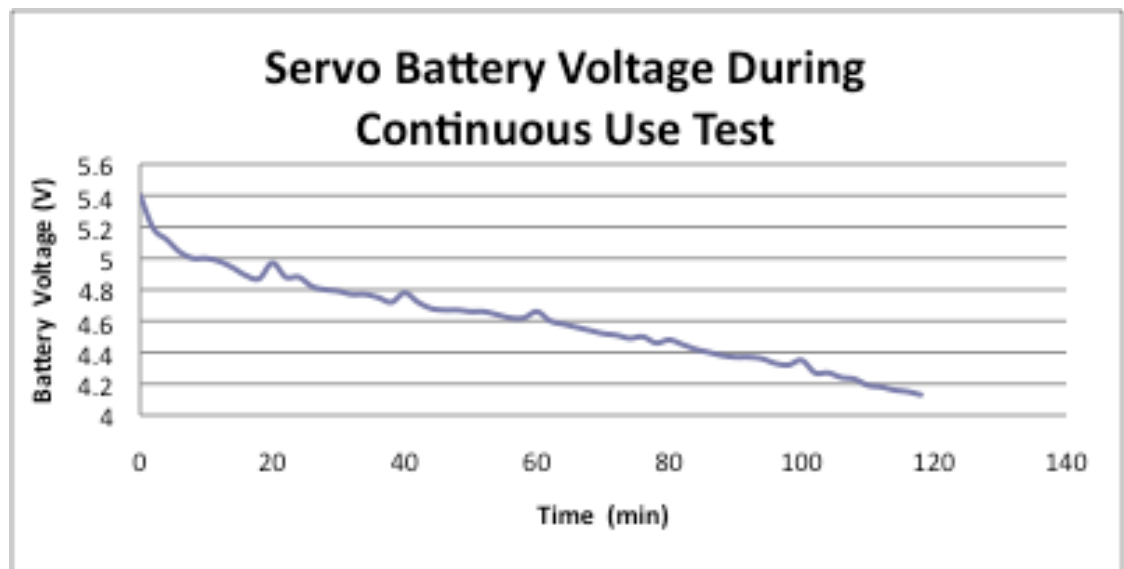


Figure 41 Decrease in servo power supply voltage during continuous cyclic contraction of the hand over 2 hours

Spikes in battery voltage were observed at 20 minute increments which correspond to 1 minute stops of the grasp macro in order to save force production data in Labchart.

Force production of the prosthetic hand deteriorated most rapidly during the first 40 minutes of use (Figure 42).

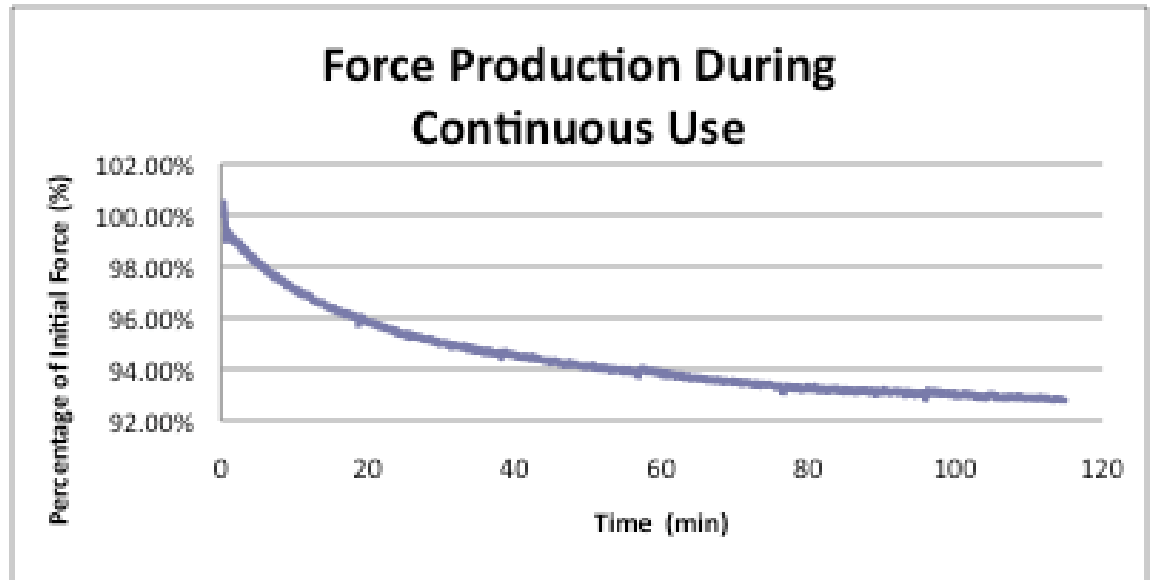


Figure 42 Decrease in grip force during continuous cyclic contraction of the hand over 2 hours

Following the first 40 minutes, grip force decreased at a rate of .025% per minute for the remaining 80 minutes or approximately 480 remaining cycles of the continuous use test. The total decline in grip force following the 120 minutes of continuous use or 720 cycles was approximately 7% of the initial grip force. Minor spikes in grip force can be observed at 20-minute intervals in response to the brief stops of the grasp macro.

6. Conclusions

The continuous use test was conducted over 120 minutes, which corresponded to approximately 720 cycles of the hand through the grasping and unloading cycle around the hand dynamometer. During this time both servo battery potential and hand force production declined most rapidly during the first 20 and 40 minutes respectively. Following these initial drops, servo battery and hand

force production decreased at relatively stable rates with force production decreasing only 7% over the 120 minutes of continuous use. This test was carried out to characterize the performance of the prototype prosthetic hand's 6V power supply which was made up of 4 AA batteries arranged in series. While this characterization demonstrated that this readily available battery form factor can perform relatively consistently over the first 2 hours of continuous use, other battery types including rechargeable lithium battery packs may provide more space efficiency and higher current output in future iterations of the prosthetic hand design.

Spherical Object Texture and Weight Effects on Grip Test

1. Purpose

The purpose of this lab is to evaluate the lifting capacity of the prosthetic hand using uniform spherical objects. While the hand can be utilized for a variety of tasks this allows for the influence of properties such as weight, diameter, and surface texture to be compared.

2. Description

Spherical objects weighted down with additional loads of various sizes will be picked up from a flat surface to objectively evaluate the grip capabilities of the prosthetic hand. Object properties including weight and diameter will be recorded for each test object to determine the optimal functional range of the prosthetic hand's current configuration. A lifting bag will be used in order to

increase the weight of each sample object to the point that the object cannot be lifted while keeping the object's outer diameter.

3. Materials and Equipment

- 3.1. Caliper
- 3.2. Scale
- 3.3. Lifting net
- 3.4. Spectra fishing line
- 3.5. Open Source Prosthetic Hand
- 3.6. Sports balls of various size
- 3.7. Painter's tape

4. Protocol

- 4.1. Record each object's weight, diameter, surface texture, and description (Figure 43)



Figure 43 spherical objects used to characterize the hand's lifting properties: a Baseball, roller hockey ball, tennis ball, golf ball, and ping pong ball

- 4.2. Attach the lifting net to the spherical object by securing the loop of Spectra fishing line to the top of the first ball to be lifted with painters tape (Figure 44)

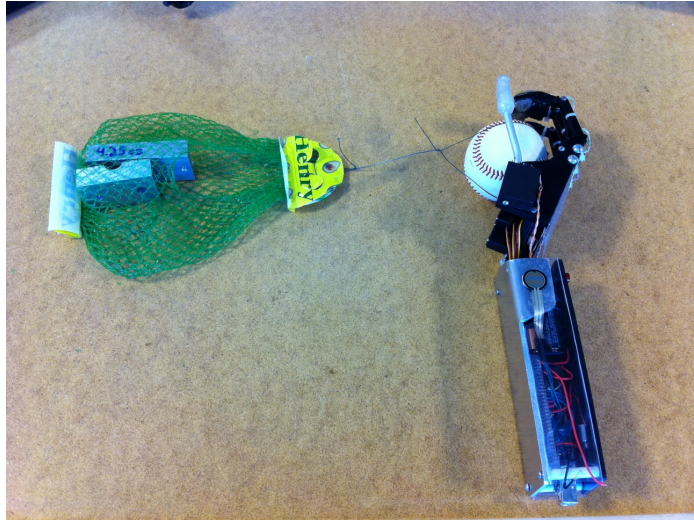


Figure 44 The lifting net loaded with a known mass attached to a spherical object used to characterize the hand's lifting properties. The purpose of the net was to increase the load lifted with each ball without changing the object's outer diameter or texture.

- 4.3. Place the spherical object on a flat surface
- 4.4. Attempt to lift the object and lifting net completely off the table using the prosthetic hand, hold the object for 15 seconds and release the object
- 4.5. If the object is not successfully lifted reposition and repeat lifting up to 3 times
- 4.6. Record any failures or deviations from protocol
- 4.7. Repeat steps 2-6 for all test objects

5. Results

Lifting capacity of the open source prosthetic hand was greatest for objects of smaller diameter with the smallest object, the ping pong ball, allowing the hand to grasp up to 13.375 oz loaded beneath the ball (Figure 45).

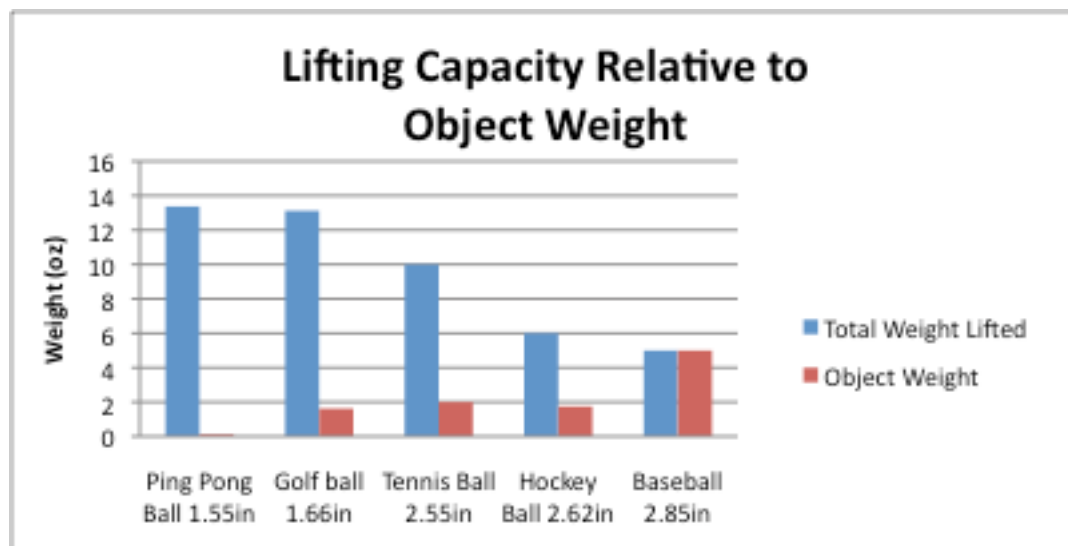


Figure 45 Lifting capacity of the hand with respect to each object's weight and diameter. The hand was capable of lifting greater total loads as the diameter of the object decreased

6. Conclusions

The spherical object's texture and weight effects on grip test demonstrated that the hand's grasping strength is greatest for smaller diameter objects. This finding is consistent with assumptions made during the design of the fingers which suggested that the levers which cause the "auto-curling" motion of the fingers would have the greatest grip force as the fingers reach their most closed point with the weakest point grasp force occurring at the hand's most open position. This can be likened to ease of gripping a golf ball compared to palming a basketball.

Limited Real-World Use Assessment

The open source prosthetic hand was also tested in a variety of everyday tasks including, sorting small objects, grasping and moving objects less than 5lbs, dialing on a digital key pad, and other miscellaneous tasks. This characterization found that the hand was effective at manipulating both small and delicate items such as an egg in the one finger “pinch” mode as well as lifting heavier objects such as a water bottle in the three finger “grasp” mode (Figure 46).

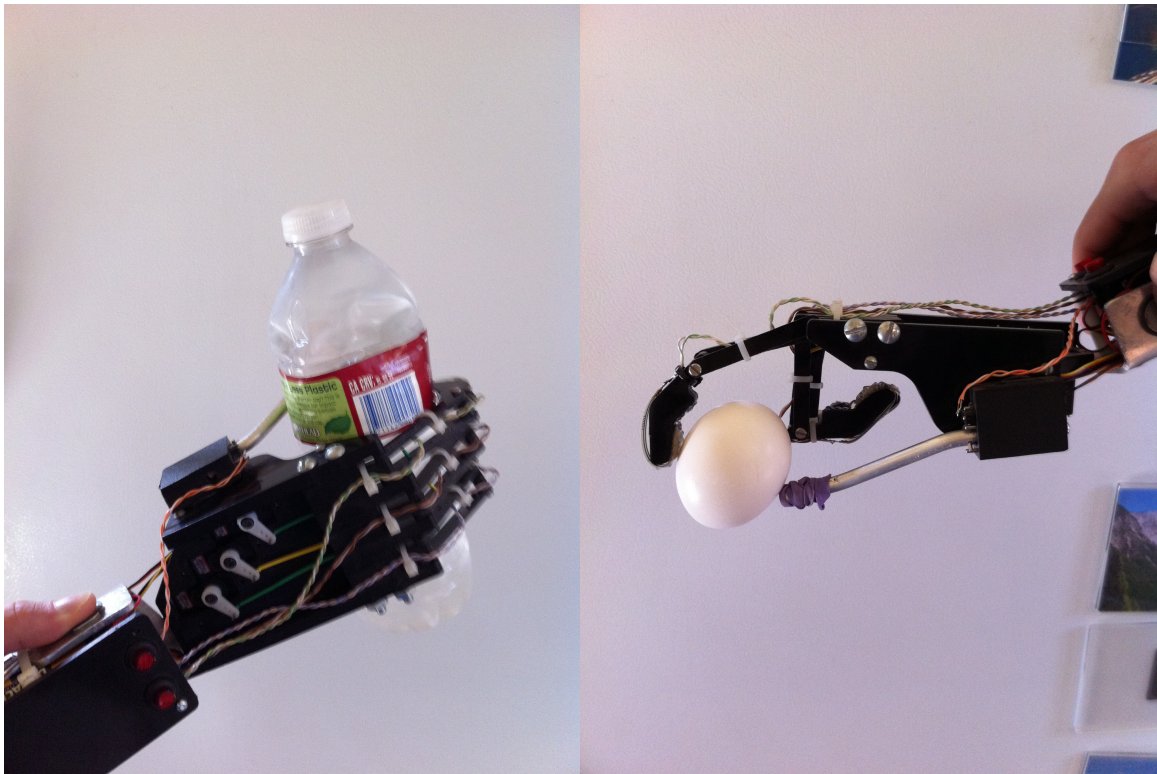


Figure 46 Lifting of both heavy and delicate objects using the prosthetic hand’s “grasp” and pinch modes

This range of basic functionality demonstrates that the open source prosthetic hand platform could be utilized as an effective supplement to a less dexterous but more ruggedized prosthetic such as a body powered mechanical prosthetic hand.

Myoelectric Input Functional Test

The open source prosthetic hand was also tested using a myoelectric input designed and fabricated by Nickolas Butler at Cal Poly San Luis Obispo [16] (Figure 47).

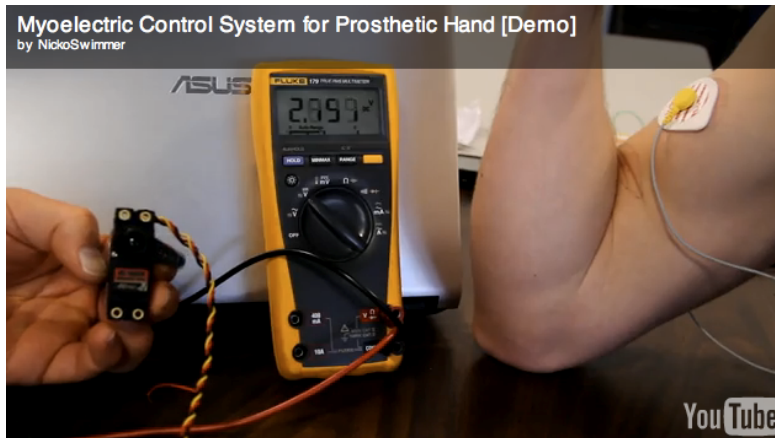


Figure 47 Myoelectric circuit configured to read muscle action potentials in the arm [16]

This circuit is capable of reading the myoelectric signal produced by muscles in the arm relative to neutral ground on the body. The raw EMG signal is converted to a 0 to 5 volt signal which will be read by the Arduino open source prosthetic hand. This mode of actuation was tested for a variety of functional tests utilizing the hand's grasp and pinch modes. Initial testing found that with minimal changes to the open source prosthetic hand's firmware, the myoelectric input could be utilized to complete all of the tasks which were originally coordinated with the hand's original FSR input. Due to the erratic nature of EMG signal the motion of the fingers was fairly jittery at positions in between a fully open and closed grasp. Further optimization of the hand's firmware could be utilized to condition the myoelectric input and smoothen the resulting motion of the fingers.

6.0 CONCLUSIONS

The open source prosthetic hand developed for this thesis delivers some of the more dexterous motions of available robotic prosthetics at a significantly lower cost. This platform could provide a foundation for more affordable electronic prosthetic hands which more closely mimic native hand function and appearance at a price point which could make these devices more accessible to a wider user base.

While significant research has gone into the development of extremely dexterous robotic prosthetics, these devices remain prohibitively expensive with commercially available models such as the i-LIMB costing approximately \$20,000 before insurance coverage and emerging models such as the DEKA “Luke” arm approaching \$100,000. The Arduino open source microcontroller provides an ideal platform to deliver the advanced signal conditioning and functionality of more expensive alternatives at reduced cost while also maximizing the potential for customization of robotic prosthetics for a user’s unique needs.

Testing of the prosthetic hand developed for this thesis demonstrated the device’s capability to provide consistent grasping strength during up to two hours of continuous use. While this is promising, the platform could also be configured with higher capacity rechargeable lithium ion batteries which could enhance battery life during sustained use and high intensity activity. Also the hand performed well in handling various spherical objects up to approximately 14 oz within the device’s grasp range. This range of performance was sufficient to

complete various everyday tasks including manipulating and sorting small objects, opening doors, grasping moderately heavy objects such as water bottles, and sensitive objects such as an egg. During these actions the hand's force feedback sensors in the distal tips were able to provide a variable sensory output depending on the utilization of the distal tip of the fingers.

While the prosthetic hand platform was capable of demonstrating dexterous functionality, several additional areas of the hand's design could be addressed in future work. Consolidating the electronics to a single printed circuit board would help to improve space efficiency and reduce the probability of electrical malfunction. Also utilization of a single higher capacity rechargeable lithium ion power supply could improve battery life and grasp strength performance during sustained use. Research into additional force sensitive inputs from sensors on the fingers could enhance the user's dexterity and perception of the grip conformity on the hand. These force sensors could utilize multiple force sensors in arrays which cover a larger portion of each finger's exposed grasping area or could provide feedback regarding the position on each finger where grip force is being applied the most. The hand's push-pull rod based actuation system could be optimized to reduce rod deflection in order to maximize flexion and extension of the fingers during use and minimize the internal resistance to these motions. Ways to reduce this resistance could include more direct routing paths from the servo heads to the finger attachment points which would allow more ridged rods to be used. Furthermore once the electronics were reduced to their minimal profile, a smaller electronics enclosure

could be developed which would more readily integrate with a patient's residual limb socket. This enclosure could incorporate easy battery exchange and charging in addition to creating a universal connection for pressure, myoelectric, and other sensors used to control actuation of the hand. The hand could be further improved by creating a silicone covering which could provide protection of the hand's joints and electrical components while enhancing the life-like appearance of the prosthetic.

Additional functional modifications could be made through further research in the signal conditioning and firmware. These modifications could include onboard conditioning of amplified myoelectric signal, integration with more advanced force sensing arrays in the fingers, provide finger specific feedback of force sensing using multiple vibrating motors or electrical stimulating outputs, and development of additional algorithms to improve finger specific grip to irregular shaped objects based on finger force sensing inputs.

While the final prototype of the open source prosthetic hand platform succeeded in accomplishing its primary goal of providing dexterous functionality at relatively low cost, the current iteration of the platform still has several limitations. As mentioned previously the electrical components of the final prototype could be miniaturized and mounted on a single board in order to improve the reliability and space efficiency. Also, further development could focus on packaging of the hand so that the joints and electrical components would be less vulnerable to fouling from liquids and debris. Lastly integration of a

higher charge density battery and refinement of the finger push rod system, could enhance grip force and increase the load capacity of the hand during daily use.

Future iterations of this open source prosthetic hand platform could provide a prosthetic hand which utilizes inexpensive off the shelf electrical components to provide the advanced functionality of more expensive commercially available robotic hand prosthetic devices. By making robotic prosthetic devices available to a wider user base, patients will not have to compromise exclusively based on the most affordable and basic prosthetic available but will instead have access to a platform which can recreate many of dexterous functions and appearance of the users native hand.

WORKS CITED

- [1] LV McFarland•; SLH Winkler•; AW Heinemann•; M Jones•; A Esquenazi•;, “Unilateral upper-limb loss•: satisfaction and prosthetic-device use in veterans and servicemembers from Vietnam and OIF/OEF conflicts,” *Journal of rehabilitation research and development*, vol. 47, no. 4, pp. 299-316, 2010.
- [2] “Combination of a total free forearm flap and a sensate local flap for preservation of the shoulder girdle in massive, nonreplantable upper extremity injuries,” *The Journal of trauma*, vol. 66, no. 2, pp. 561-563.
- [3] H Bouwsema•; CK van der Sluis•; RM Bongers•;, “Movement Characteristics of Upper Extremity Prostheses During Basic Goal-directed Tasks,” *Clinical biomechanics*, vol. 25, no. 6, pp. 523-529, 2010.
- [4] C Toledo•; A Vera•; L Leija•; A Ram7;rez•; R Mu1;oz, “Upper Limb Prostheses for Amputations Above Elbow: A Review,” *2009 Pan American Health Care Exchanges*, pp. 104-108, 2009.
- [5] Maurice LeBlanc, “MAKING THE CASE FOR BODY-POWERED UPPER-LIMB PROSTHESES,” *RESNA-Assoc for the Advancement of Rehabilitation Technology*, pp. 196-198.
- [6] Nan Li•; Xinqing Wang•; Hong Liu•; Li Jiang•; Shaowei Fan•; Dapeng Yang, “Development of an Anthropomorphic Prosthetic Hand for Man-machine Interaction,” *Lecture notes in computer science*, vol. 6424, pp. 38-46, 2010.
- [7] Kevin Carroll, “Spotlight on Technology: Pick Six,” *Amputee Golfer Magazine*, pp. 31-33, 2009.
- [8] “Touch Bionics.” [Online]. Available: <http://www.touchbionics.com/i-LIMB/grip-patterns>. [Accessed: 02-Mar-2011].
- [9] Stephanie Z Pavlou, “REVOLUTIONARY Prosthetic Arm Advances,” *O & P Business News•: Linking the Orthotic and Prosthetic Profession*, vol. 19, no. 17, pp. 24-27, 2009.
- [10] “Arduino - HomePage.” [Online]. Available: <http://arduino.cc/>. [Accessed: 01-Mar-2011].
- [11] M. McRoberts, *Beginning Arduino*. 2010.
- [12] “FSR Force Sensing Resistor Integration Guide and Evaluation Parts Catalog.” Interlink Electronics.
- [13] J Mazurkiewicz, “Advantages of Servos,” *Proceedings Electrical Insulation Conference and Electrical Manufacturing Expo*, 2005.
- [14] C Castellini•; AE Fiorilla•; G Sandini•;, “Multi-subject/daily-life Activity EMG-based Control of Mechanical Hands,” *Journal of neuroengineering & rehabilitation*, vol. 6, p. 41.
- [15] Roberto Merletti•; Enrico Merlo•; Alberto Botter•; Marco Alessandro Minetto•; Amedeo Troiano, “Technology and Instrumentation for Detection and Conditioning of the Surface Electromyographic Signal: State of the Art,” *Clinical biomechanics*, vol. 24, no. 2, pp. 122-134, 2009.

- [16] N. Butler, "Development of a Miniturized Myoelectric Detection for General Purpose Human-Machine Interface Applications," 2011.

APPENDIX A: CODE

This appendix details the Arduino code modules used to validate individual subcomponents of the open source prosthetic hand platform. These modules include algorithms for controlling servo motors (ServoRock, and ServoRock2), reading and scaling input from a force sensitive resistor (FSR), reading and scaling input from a current sensor (CurrentSensingModule), and reading tilt and orientation data from an accelerometer (Accelerometer Sensor).

ServoRock

```
//ServoRock
#include <Servo.h>

int val;
Servo myservo;

void setup() {
  myservo.attach(9);
}

void loop() {

  for (int i=15; i <= 165; i++){
    myservo.write(i);
    delay(10);
  }
  delay(1000);

  for (int i=165; i >= 15; i--){
    myservo.write(i);
    delay(10);
  }
  delay(1000);
}
```


ServoRock2

```
//ServoRock2
#include <Servo.h>

int val;
Servo servo1;
Servo servo2;

void setup() {
  servo1.attach(9);
  servo2.attach(10);
}

void loop() {

  for (int i=15; i <= 165; i++){
    servo1.write(i);
    delay(1);
  }
  for (int i=15; i <= 165; i++){
    servo2.write(i);
    delay(1);
  }
  delay(1000);

  for (int i=165; i >= 15; i--){
    servo1.write(i);
    delay(1);
  }
  for (int i=165; i >= 15; i--){
    servo2.write(i);
    delay(1);
  }
  delay(1000);
}
```

FSR

```
//FSR
int FSR = 0;    // select the input pin for the LDR
int val=0;

void setup(){
  Serial.begin(9600);    // open the serial port at 9600 bps:
}
```

```

void loop() {

    val = analogRead(FSR);    // read the value from the sensor
    Serial.println(val);
    delay (100);
}

```

CurrentSensingModule

```

//Current sensing module
#include <Servo.h>
Servo SERVO1;

int FSR = 0;    // select the input pin for the LDR
int val=0;

void setup(){
    SERVO1.attach(10);
    Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {
    SERVO1.write(30);
    val = analogRead(FSR);    // read the value from the sensor
    Serial.println(val);
    delay (1000);
}

```

Accelerometer Sensor

```

// Accelerometer Sensor http://www.arduino.cc/en/Tutorial/Memsic2125
const int xPin = 2;
const int yPin = 3;

void setup(){
    Serial.begin(9600);
    pinMode(xPin,INPUT);
    pinMode(yPin, INPUT);
}

void loop(){
    int pulseX, pulseY;
    int accelerationX, accelerationY;
    pulseX=pulseIn(xPin,HIGH);
    pulseY=pulseIn(yPin,HIGH);

    accelerationX = ((pulseX /10) -500)*8;
    accelerationY = ((pulseY /10) -500)*8;
}

```

```
Serial.print(accelerationX);  
Serial.print("\t");  
Serial.print(accelerationY);  
Serial.println();  
delay(100);  
}
```

APPENDIX B: CODE

This appendix includes the Arduino code algorithms used for various stages of the open source prosthetic hand's development. These algorithms include, the proof of concept prototype code (Open Hand 1.0) through the final prototype (Open Hand 1.3.1).

Proof of concept prototype Code:

```
//OpenHand1.0
```

```
//Inputs:
```

```
//FSR(s)
```

```
int FSR1 = 0;    // variable to store the value coming from the sensor
```

```
//Button(s)
```

```
const int buttonPin = 2;  // the pin that the pushbutton is attached to
```

```
int buttonPushCounter = 0; // counter for the number of button presses
```

```
int buttonState = 0;      // current state of the button
```

```
int lastButtonState = 0;  // previous state of the button
```

```
//Accelerometer
```

```
const int xPin = 7;
```

```
const int yPin = 8;
```

```
//Outputs:
```

```
//Servo(s)
```

```
#include <Servo.h>
```

```
Servo SERVO1;
```

```
Servo SERVO2;
```

```
Servo SERVO3;
```

```
Servo SERVO4;
```

```
int ServoVal1 = 0;
```

```
int ServoVal2 = 0;
```

```
int ServoVal3 = 0;
```

```
int ServoVal4 = 0;
```

```
//VibrationMotor(s)
```

```
int Vib1 = 0;
```

```

void setup() {
//Inputs:
  //Accelerometer
  pinMode(xPin,INPUT);
  pinMode(yPin, INPUT);

//Outputs:
  //Servos
  SERVO1.attach(6);
  SERVO2.attach(9);
  SERVO3.attach(10);
  SERVO4.attach(11);

  //Communication
  Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {
//Inputs
  //FSR
  FSR1 = analogRead(0);    // read the value from the sensor

  //Button state change detection and set # of cases
  buttonState = digitalRead(buttonPin);
  if (buttonState != lastButtonState) {
    if (buttonState == HIGH) {
      buttonPushCounter++;
    }
    else {
    }
    lastButtonState = buttonState;
    if(buttonPushCounter > 3){
      buttonPushCounter=0;
    }
  }
  delay (10);

  //Case 1: Hand opens completely no input control
  if (buttonPushCounter == 0){
    //Servo(s)
    SERVO1.write(120);
    SERVO2.write(120);
    SERVO3.write(120);
    SERVO4.write(120);
  }
}

```

```

    //Communication
    Serial.println(FSR1);
}

//Case 2: FSR control (All)
else if ( buttonPushCounter == 1){
    //Servos
    ServoVal1 = map(FSR1, 0, 133, 10, 170);    //Interpolates FSR value to
corresponding servo position
    ServoVal2 = map(FSR1, 0, 950, 10, 170);
    ServoVal3 = map(FSR1, 0, 950, 10, 170);
    ServoVal4 = map(FSR1, 0, 950, 10, 170);
    //Communication
    Serial.println(ServoVal1);
}

//Case 3: FSR control (Pinch)
else if ( buttonPushCounter == 2){
    //Servos
    ServoVal1 = map(FSR1, 0, 950, 10, 170);    //Interpolates FSR value to
corresponding servo position
    ServoVal2 = 60;
    ServoVal3 = 60;
    ServoVal4 = 60;
    //Communication
    Serial.println(3);
}

//Case 4: Accelerometer control (All)
else if ( buttonPushCounter == 3){
    //Accelerometers
    int pulseX, pulseY;
    int accelerationX, accelerationY;
    pulseX=pulseIn(xPin,HIGH);
    pulseY=pulseIn(yPin,HIGH);
    accelerationX = ((pulseX /10) -500)*8;
    accelerationY = ((pulseY /10) -500)*8;
    //Servos

    ServoVal1 = map(accelerationY, 0, 1000, 10, 170);    //Interpolates FSR
value to corresponding servo position
    ServoVal2 = map(FSR1, 0, 950, 10, 170);
    ServoVal3 = map(FSR1, 0, 950, 10, 170);
    ServoVal4 = map(FSR1, 0, 950, 10, 170);

    //Communication

```

```

    Serial.println(accelerationY);
}

//Case 5: FSR(s) decrease Grip Force (maybe another switch can toggle
between light hard and soft grips while control signal opens hand)

//Case Independent Functions
constrain (ServoVal1, 10, 170); //Constrains possible servo position values
constrain (ServoVal2, 10, 170);
constrain (ServoVal3, 10, 170);
constrain (ServoVal4, 10, 170);

SERVO1.write(ServoVal1); //Writes position value to servo
SERVO2.write(ServoVal2);
SERVO3.write(ServoVal3);
SERVO4.write(ServoVal4);

//VibrationMotors
Vib1 = map(ServoVal1, 10, 170, 0, 150);
constrain (Vib1, 0, 150);
analogWrite(6,Vib1);

//Debug
//Serial.println(ServoVal1);
}

```

Open Hand 1.1.0 Code:

```

//OpenHand1.1

//Inputs:
//FSR(s)
int FSR1 = 0;    // variable to store the value coming from the sensor

//Button(s)
const int  buttonPin = 2;  // the pin that the pushbutton is attached to
int buttonState = 0;      // current state of the button

//Outputs:
//Servo(s)
#include <Servo.h>
Servo SERVO1;
Servo SERVO2;
Servo SERVO3;
Servo SERVO4;

```

```

int ServoVal1 = 0;
int ServoVal2 = 0;
int ServoVal3 = 0;
int ServoVal4 = 0;

//VibrationMotor(s)
int Vib1 = 0;

void setup() {

//Outputs:
//Servos
SERVO1.attach(9);
SERVO2.attach(10);
SERVO3.attach(11);
//SERVO4.attach(11);

//Communication
Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {
//Inputs
//FSR
FSR1 = analogRead(0);    // read the value from the sensor

//Thumb Sensor Switch
buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {
//Servos
ServoVal1 = map(FSR1, 0, 133, 10, 170);    //Interpolates FSR value to
corresponding servo position
ServoVal2 = 80;
ServoVal3 = 80;
//ServoVal4 = 80;
//Communication
//Serial.println("On");
}
else {
//Servos
ServoVal1 = map(FSR1, 0, 133, 10, 170);    //Interpolates FSR value to
corresponding servo position
ServoVal2 = map(FSR1, 0, 133, 10, 170);
ServoVal3 = map(FSR1, 0, 133, 10, 170);
//ServoVal4 = map(FSR1, 0, 133, 10, 170);
//Communication

```



```

    //Serial.println("OFF");
}

//Case Independent Functions
constrain (ServoVal1, 10, 100); //Constrains possible servo position values
constrain (ServoVal2, 10, 100);
constrain (ServoVal3, 10, 100);
//constrain (ServoVal4, 10, 100);

SERVO1.write(ServoVal1); //Writes position value to servo
SERVO2.write(ServoVal2);
SERVO3.write(ServoVal3);
//SERVO4.write(ServoVal4);

//VibrationMotors
Vib1 = map(ServoVal1, 10, 170, 0, 200);
constrain (Vib1, 0, 200);
analogWrite(6,Vib1);

//Debug
//Serial.println(ServoVal1);
}

```

Open Hand 1.1.1 Code:

```

//OpenHand1.1.1

//Inputs:
//FSR(s)
int FSR1 = 0; // variable to store the value coming from the sensor
int ServoCurrent1 = 0;
//Button(s)
const int buttonPin = 2; // the pin that the pushbutton is attached to
int buttonState = 0; // current state of the button

//Outputs:
//Servo(s)
#include <Servo.h>
Servo SERVO1;
Servo SERVO2;
Servo SERVO3;

int ServoVal1 = 0;
int ServoVal2 = 0;
int ServoVal3 = 0;

```

```

//VibrationMotor(s)
int Vib1 = 0;

void setup() {

//Outputs:
//Servos
SERVO1.attach(9);
SERVO2.attach(10);
SERVO3.attach(11);
//SERVO4.attach(11);

//Communication
Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {
  //Inputs
  //FSR
  FSR1 = analogRead(0);    // read the value from the sensor
  //Servo Current Draw
  ServoCurrent1 = analogRead(1);
  Serial.println(ServoCurrent1);

  //Thumb Sensor Switch
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    //Servos
    ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
corresponding servo position
    ServoVal2 = 80;
    ServoVal3 = 80;

    //Communication
    //Serial.println(ServoCurrent1);
  }
  else {
    //Servos
    ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
corresponding servo position
    ServoVal2 = map(FSR1, 0, 250, 10, 170);
    ServoVal3 = map(FSR1, 0, 250, 10, 170);

    //Communication
    //Serial.println("OFF");
  }
}

```

```

}

//Case Independent Functions
constrain (ServoVal1, 10, 100); //Constrains possible servo position values
constrain (ServoVal2, 10, 100);
constrain (ServoVal3, 10, 100);

SERVO1.write(ServoVal1); //Writes position value to servo
SERVO2.write(ServoVal2);
SERVO3.write(ServoVal3);

//VibrationMotors
Vib1 = map(ServoVal1, 10, 170, 0, 200);
constrain (Vib1, 0, 200);
analogWrite(6,Vib1);

//Debug
//Serial.println(ServoVal1);
}

```

Open Hand 1.2.1 Code:

```

//OpenHand1.2.1

//Inputs:
//FSR(s)
int FSR1 = 0; // variable to store the value coming from the sensor
int FSR2 = 0;
int FSR3 = 0;
int FSR4 = 0;
//Button(s)
const int buttonPin = 2; // the pin that the pushbutton is attached to
int buttonState = 0; // current state of the button

//Outputs:
//Servo(s)
#include <Servo.h>
Servo SERVO1;
Servo SERVO2;
Servo SERVO3;

int ServoVal1 = 0;
int ServoVal2 = 0;
int ServoVal3 = 0;

```

```

//VibrationMotor(s)
int VibSum = 0;
int Vib1 = 0;

void setup() {

//Outputs:
//Servos
SERVO1.attach(9);
SERVO2.attach(10);
SERVO3.attach(11);
//SERVO4.attach(11);

//Communication
Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {
//Inputs
//FSR
FSR1 = analogRead(0);    // read the value from the sensor
FSR2 = analogRead(1);    // read the value from the sensor
FSR3 = analogRead(2);    // read the value from the sensor
FSR4 = analogRead(3);    // read the value from the sensor
//Serial.println(FSR2);

//Thumb Sensor Switch
buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {
//Servos
ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
corresponding servo position
ServoVal2 = 80;
ServoVal3 = 80;

//Communication
//Serial.println(ServoCurrent1);
}
else {
//Servos
ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
corresponding servo position
ServoVal2 = map(FSR1, 0, 250, 10, 170);
ServoVal3 = map(FSR1, 0, 250, 10, 170);
}
}

```

```

    //Communication
    //Serial.println("OFF");
}

//Case Independent Functions
ServoVal1= constrain (ServoVal1, 10, 100); //Constrains possible servo
position values
ServoVal2= constrain (ServoVal2, 10, 100);
ServoVal3= constrain (ServoVal3, 10, 100);

SERVO1.write(ServoVal1); //Writes position value to servo
// Easter Egg
if (FSR4 >200){
    SERVO2.write(10);
}
else {
    SERVO2.write(ServoVal2);
}

SERVO3.write(ServoVal3);

//VibrationMotors
VibSum = (FSR2 + FSR3 + FSR4);
Vib1 = map(VibSum, 0, 30, 0, 200);
Vib1 = constrain (Vib1, 0, 200);
analogWrite(6,Vib1);

//Debug
Serial.println(FSR4);
}

```

Open Hand 1.3.1 Code:

```

//OpenHand1.2.1

//Inputs:
//FSR(s)
int FSR1 = 0;    // variable to store the value coming from the sensor
int FSR2 = 0;
int FSR3 = 0;
int FSR4 = 0;
//Button(s)
const int buttonPin = 2; // the pin that the pushbutton is attached to

```

```

int buttonState = 0;      // current state of the button

//Outputs:
//Servo(s)
#include <Servo.h>
Servo SERVO1;
Servo SERVO2;
Servo SERVO3;

int ServoVal1 = 0;
int ServoVal2 = 0;
int ServoVal3 = 0;

//VibrationMotor(s)
int VibSum = 0;
int Vib1 = 0;

void setup() {

//Outputs:
//Servos
SERVO1.attach(9);
SERVO2.attach(10);
SERVO3.attach(11);
//SERVO4.attach(11);

//Communication
Serial.begin(9600);      // open the serial port at 9600 bps:
}

void loop() {
  //Inputs
  //FSR
  FSR1 = analogRead(0);   // read the value from the sensor
  FSR2 = analogRead(1);   // read the value from the sensor
  FSR3 = analogRead(2);   // read the value from the sensor
  FSR4 = analogRead(3);   // read the value from the sensor
  //Serial.println(FSR2);

  //Thumb Sensor Switch
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    //Servos
    ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
    corresponding servo position

```

```

ServoVal2 = 80;
ServoVal3 = 80;

//Communication
//Serial.println(ServoCurrent1);
}
else {
    //Servos
    ServoVal1 = map(FSR1, 0, 250, 10, 170);    //Interpolates FSR value to
corresponding servo position
    ServoVal2 = map(FSR1, 0, 250, 10, 170);
    ServoVal3 = map(FSR1, 0, 250, 10, 170);

    //Communication
    //Serial.println("OFF");
}

//Case Independent Functions
ServoVal1= constrain (ServoVal1, 10, 100);    //Constrains possible servo
position values
ServoVal2= constrain (ServoVal2, 10, 100);
ServoVal3= constrain (ServoVal3, 10, 100);

SERVO1.write(ServoVal1);    //Writes position value to servo
//Curling Grasp Macro activated by closing hand and pinch "pinky" finger FSR
if (FSR4 >200){
    analogWrite(6,0);
    //SERVO2.write(10);
    for(int a = 0; a < 90; a++){

        for(int i = 0; i < 80; i++){
            SERVO1.write(i);
            delay(5);
            SERVO2.write(i+10);
            delay(5);
            SERVO3.write(i+20);
            delay(5);
        }

        SERVO1.write(100);
        SERVO1.write(100);
        SERVO1.write(100);

        for(int i = 0; i < 80; i++){
            SERVO1.write(100-i);

```

```

        delay(5);
        SERVO2.write(100-(i+10));
        delay(5);
        SERVO3.write(100-(i+20));
        delay(5);
    }
}
}
else if (FSR3 > 200){
    // Full hand grasp test macro activated by closing hand and pinch middle finger
    FSR
    analogWrite(6,0);
    for(int a = 0; a < 10000; a++){
        for(int i = 0; i < 70; i++){
            SERVO1.write(i);
            SERVO2.write(i);
            SERVO3.write(i);
            delay(15);
        }
        delay(5000);

        for(int i = 0; i < 70; i++){
            SERVO1.write(90-i);
            SERVO2.write(90-i);
            SERVO3.write(90-i);
            delay(15);
        }
        delay(5000);

    }
}
else {
    SERVO2.write(ServoVal2);
}

SERVO3.write(ServoVal3);

//VibrationMotors
VibSum = (FSR2 + FSR3 + FSR4);
Vib1 = map(VibSum, 0, 30, 0, 200);
Vib1 = constrain (Vib1, 0, 200);
analogWrite(6,Vib1);

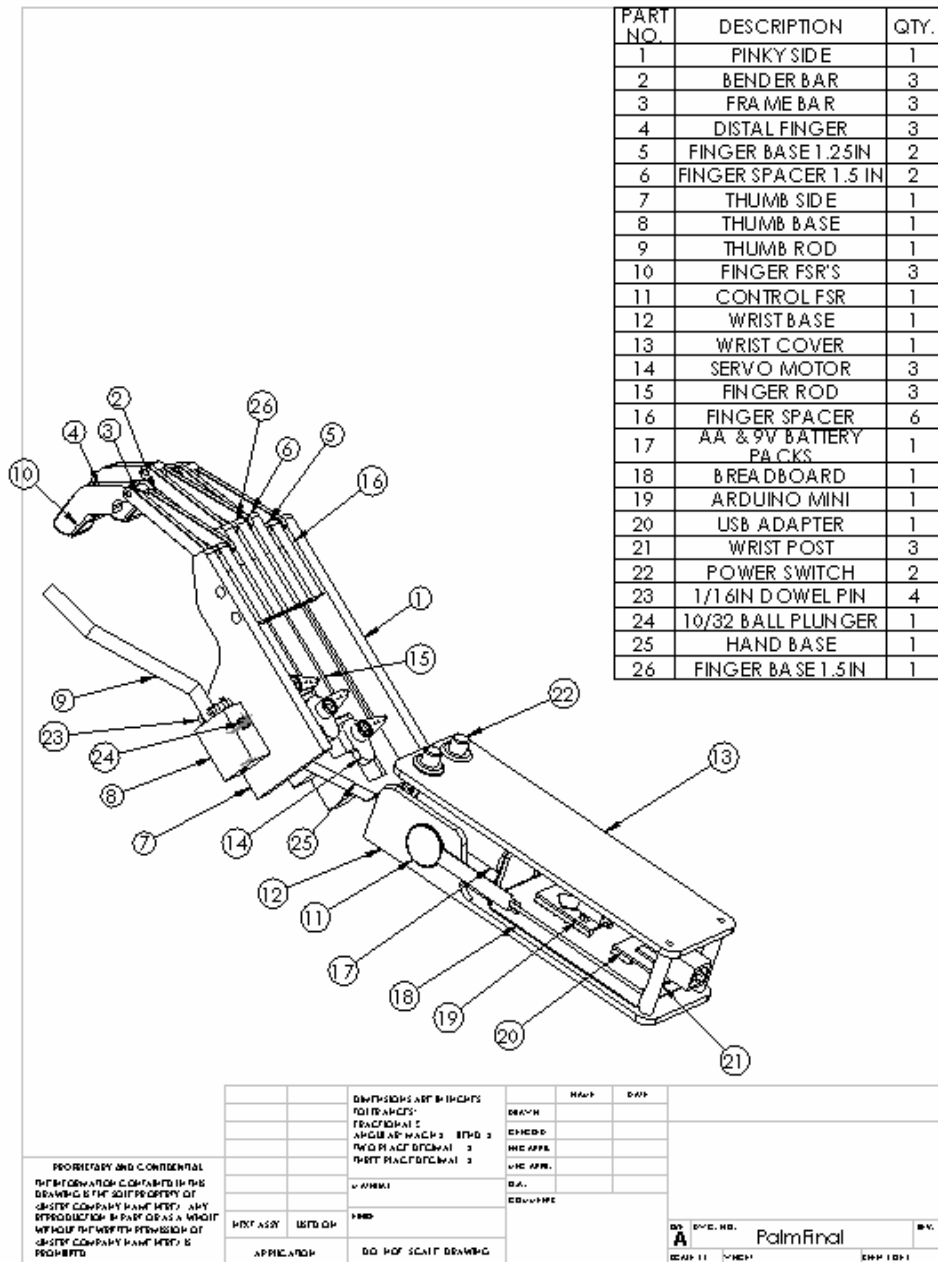
//Debug
Serial.println(FSR4);

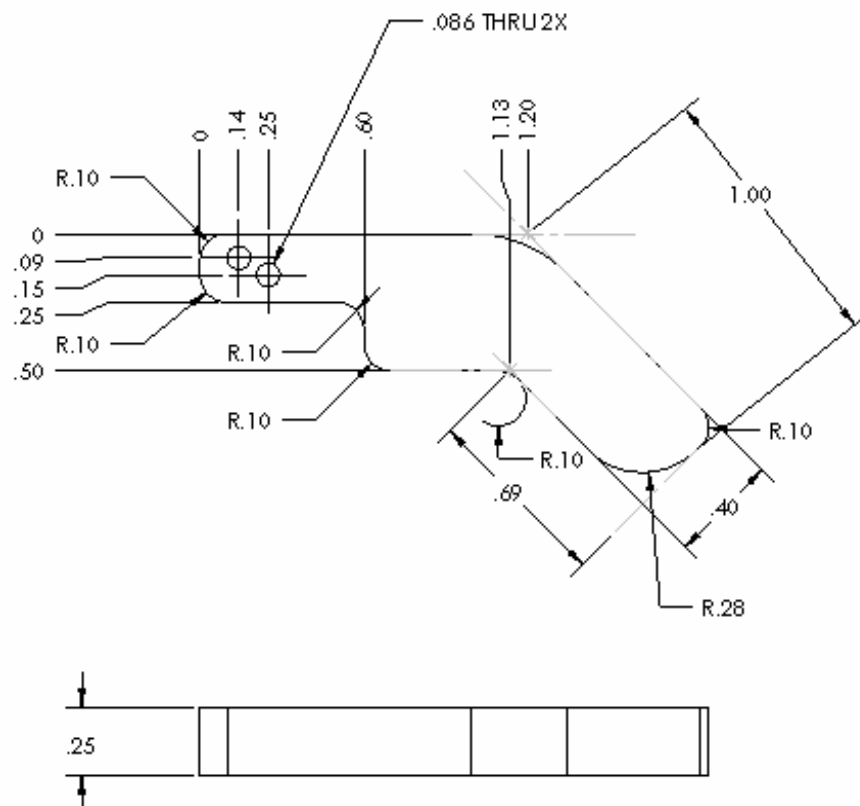
```


}

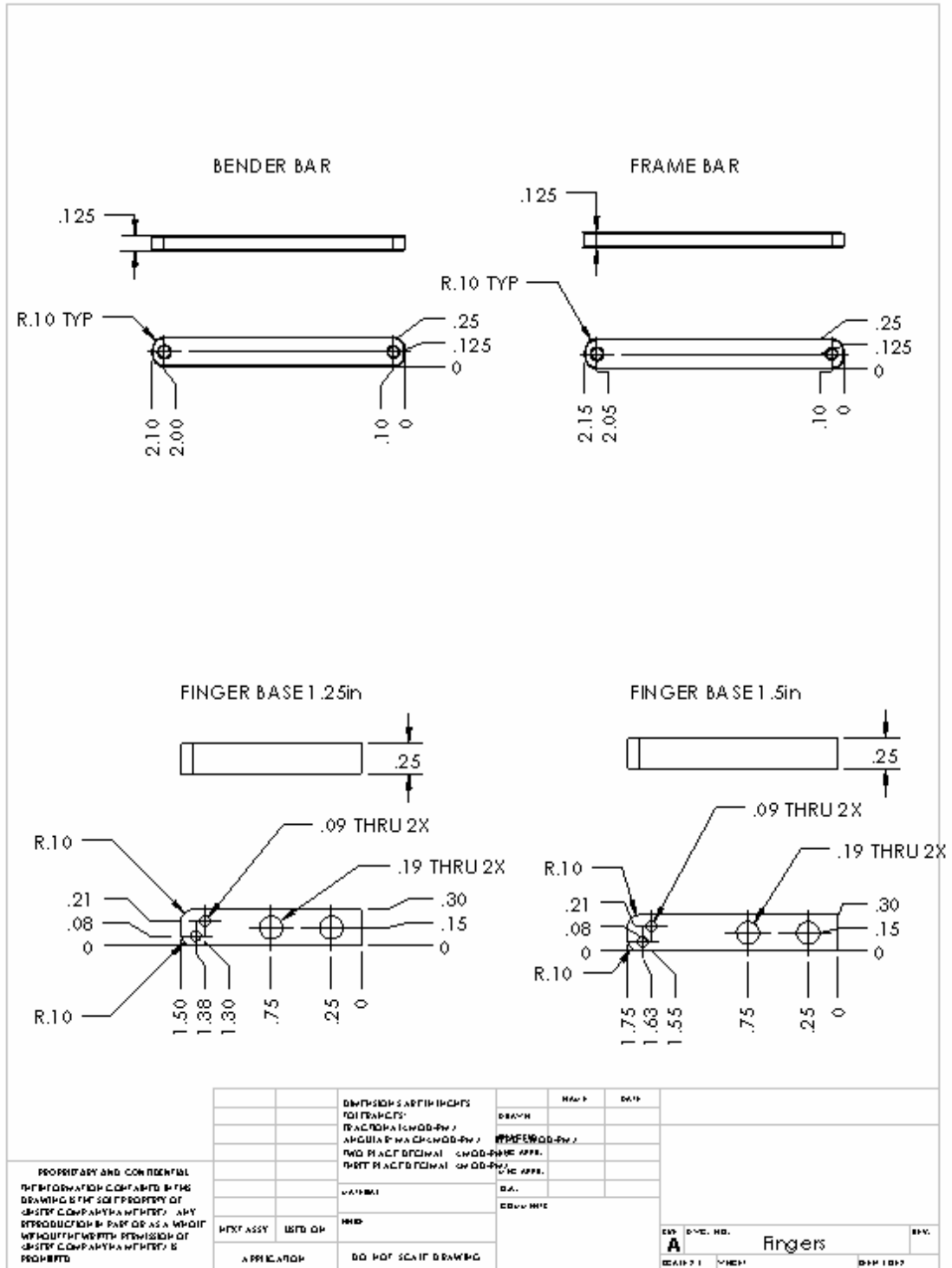
APPENDIX C: Final Prototype Engineering Drawings

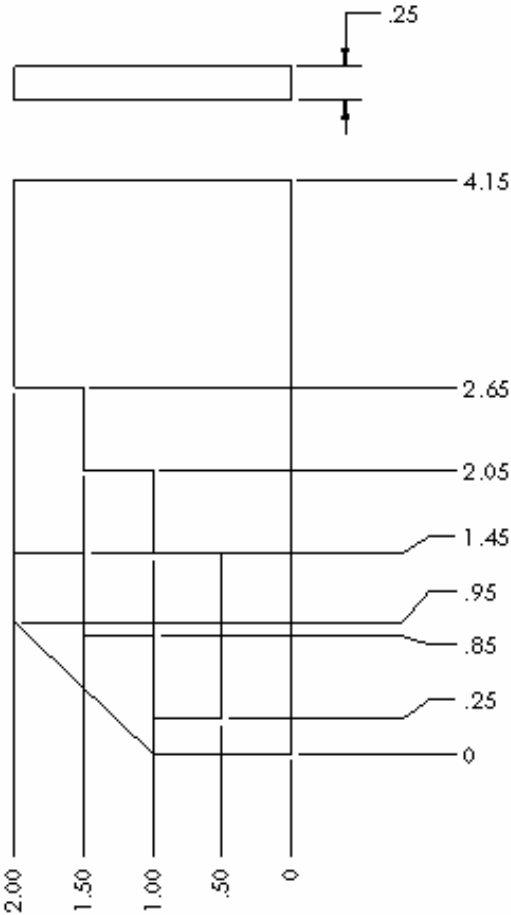
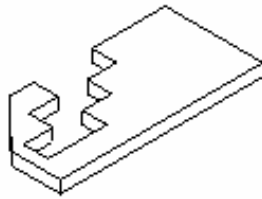
This Appendix contains Engineering drawings for the components machined to build the final prototype of the open source prosthetic hand.





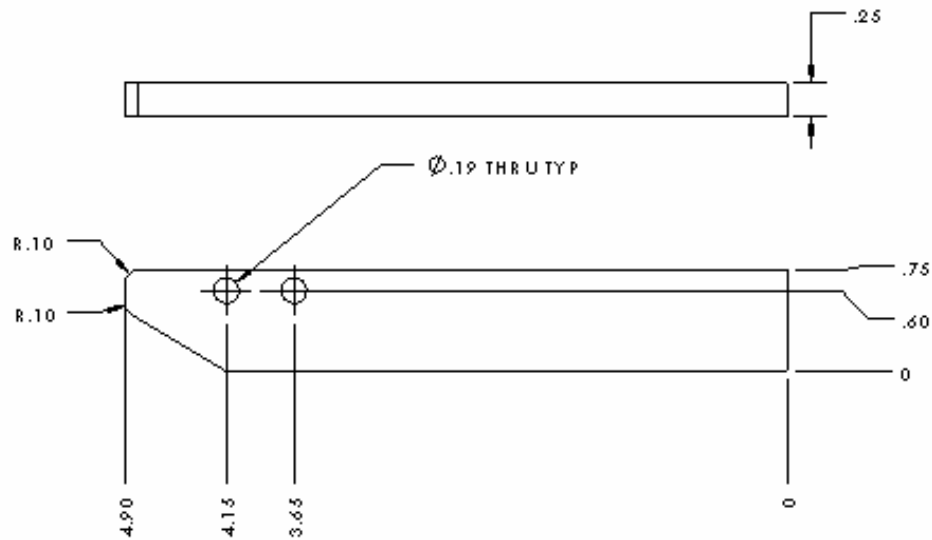
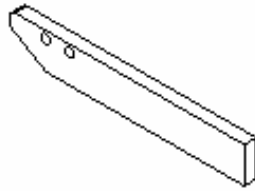
PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF JENSEN COMPANY, INC. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF JENSEN COMPANY, INC. IS PROHIBITED.		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONS .0005-.001 ANGLES .000001 DECIMALS .0005-.001 DECIMALS .0005-.001		DRAWN CHECKED DESIGNED DATE	DATE
		NEXT ASSY USED ON		PART NO. QTY. REQD.	
		APPLICATION DO NOT SCALE DRAWING		SCALE PART NO.	
		PART NO. QTY. REQD.		SCALE PART NO.	
PART NO. QTY. REQD.		PART NO. QTY. REQD.		PART NO. QTY. REQD.	



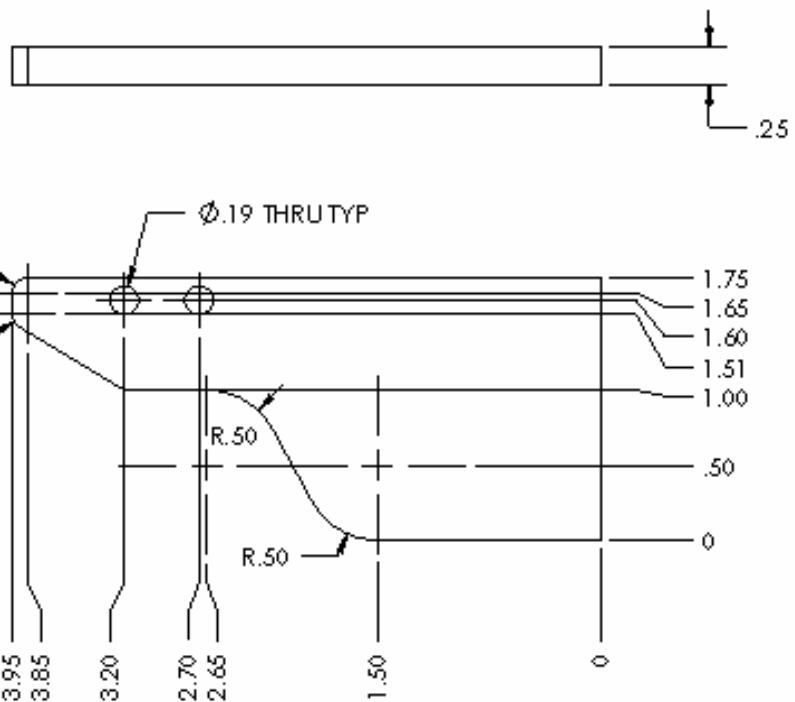


PROPERTY AND CONFIDENTIALITY INFORMATION: THIS DRAWING IS THE PROPERTY OF THE COMPANY AND IS NOT TO BE REPRODUCED OR USED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE COMPANY. ANY VIOLATION OF THIS POLICY WILL BE PROSECUTED TO THE FULL EXTENT OF THE LAW.

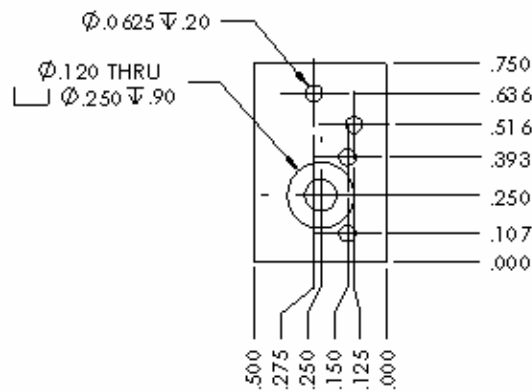
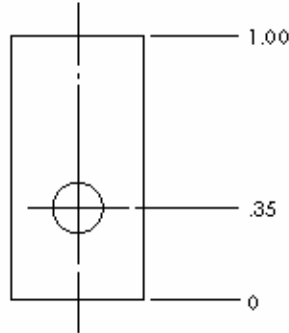
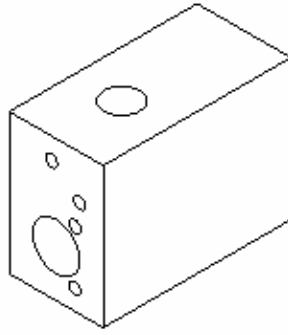
PROPERTY AND CONFIDENTIALITY INFORMATION: THIS DRAWING IS THE PROPERTY OF THE COMPANY AND IS NOT TO BE REPRODUCED OR USED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE COMPANY. ANY VIOLATION OF THIS POLICY WILL BE PROSECUTED TO THE FULL EXTENT OF THE LAW.		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONS: ± .005 DECIMALS: ± .005 ANGLES: ± .005 HOLE: ± .005 TYPICAL: ± .005		NAME: _____ DATE: _____
DESIGNER	DATE	ENGINEER	DATE	DRAWING NO. PalmDistal SHEET 1 OF 1
APPRAISER	DATE	INSPECTOR	DATE	
MANUFACTURER	DATE	FINISHER	DATE	
ASSEMBLER	DATE	TESTER	DATE	
APPLICATION		DO NOT SCALE DRAWING		DWG. NO. PalmDistal SHEET 1 OF 1



PROPERTY AND CONFIDENTIAL INFORMATION CONTAINED IN THE DRAWING IS THE PROPERTY OF THE COMPANY AND IS NOT TO BE REPRODUCED OR USED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE COMPANY.		DIMENSIONS IN INCHES FRACTIONAL AND DECIMALS THIS DRAWING IS THE PROPERTY OF THE COMPANY		NAME DATE	DRAWN CHECKED THE APPR. THE APPR. DATE
		PART NAME PART NO.		DRAWING NO.	
		APPLICATION DO NOT SCALE DRAWING		SCALE UNIT	
		PART NO.		PART NO.	
PART NAME PART NO.		PART NAME PART NO.		PART NAME PART NO.	

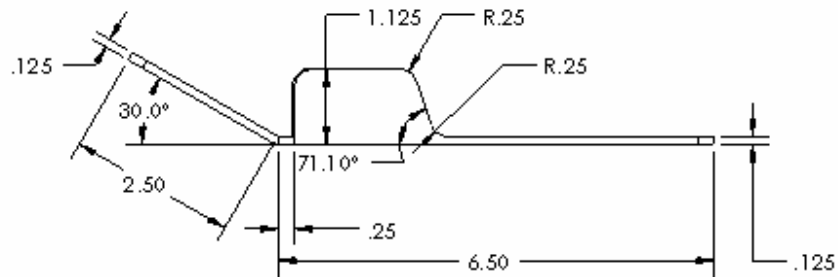


PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF CHRYSLER CREDIT CORPORATION. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF CHRYSLER CREDIT CORPORATION IS PROHIBITED.		DIMENSIONS ARE IN INCHES UNLESS OTHERWISE SPECIFIED ANGULAR DIMENSIONS IN DEGREES UNLESS OTHERWISE SPECIFIED SURFACE FINISHES UNLESS OTHERWISE SPECIFIED	DRAWN CHECKED INCH APPR. DEC APPR.	NAME DATE
	TEXT ONLY	USED ON	DIMENSIONS FINISHES MATERIALS TOLERANCES THREADS COORDINATES	
	APPLICATION	DO NOT SCALE DRAWING		



PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS
DRAWING IS THE SOLE PROPERTY OF
JENSEN COMPANY. ANY REPRODUCTION
OR USE OF THIS INFORMATION WITHOUT
THE WRITTEN PERMISSION OF
JENSEN COMPANY IS PROHIBITED.

DRAWING		DATE	BY
DESIGNED			
ENGINEERED			
MANUFACTURED			
INSPECTED			
APPROVED			
DATE			
BY			
REVISIONS			
REV.	DATE	DESCRIPTION	BY
1		Thumb mount	

95